# Travail de fin d'études: "On the gains of using autonomous trucks for freight transport."

**Auteur :** Volvert, Martin
**Promoteur(s) :** Pironet, Thierry
**Faculté :** HEC-Ecole de gestion de l'Université de Liège
**Diplôme :** Master en sciences de gestion, à finalité spécialisée en management général (Horaire décalé)
**Année académique :** 2019-2020
**URI/URL :** http://hdl.handle.net/2268.2/10157

# On the gains of using autonomous trucks for freight transport

Promoteur :
Pr Thierry PIRONET

Lectrices :
Pr Célia PAQUAY
Anaïs LEMAIRE

Travail de fin d'études présenté par
**Martin VOLVERT**
en vue de l'obtention du diplôme de
Master en sciences de gestion à finalité spécialisée en management général

Année académique 2019-2020

# On the gains of using autonomous trucks for freight transport

# Remerciements

Tchu ti, qu'est-ce qui m'a pris de reprendre un master en cours du soir? Mais après deux longues années, nous voilà enfin au bout!

Merci au Pr Pironet pour son suivi tout au long de ce travail ainsi qu'au Pr Paquay et à Mme Lemaire pour avoir accepté la lecture de ce travail.

Merci à Sophie pour sa relecture de l'Anglais. Merci aussi à Camille pour avoir su faire preuve de patience durant ces deux longues années. J'ai mes samedis de libre, maintenant!

Merci au groupe d'ingés sans qui ces deux années n'auraient peut-être pas été supportables. Aller en cours après une journée de boulot ou bien le samedi matin, ce n'est pas toujours évident, mais c'est plus facile en groupe que seul. J'attends ce resto de fin de Master!

Merci à la vague de chaleur ainsi qu'à Linkin Park, dont les chansons sont comme le bon vin et se bonifient avec l'âge, pour m'avoir accompagné durant cette rédaction.

Merci à l'ULiège d'avoir réduit le taux d'inscription de ma deuxième année de master.

# On the gains of using autonomous trucks for freight transport

Martin Volvert

**Abstract**

The world is on the edge of an economical breakthrough with the emergence of autonomous cars. Such vehicles do not, or for a very short time, require the driver's attention or even not require a driver at all. This new era of vehicles is largely promoted by Tesla's eccentric CEO Elon Musk and the promotion of the S 3 X Y series of vehicles which include an autopilot mode. In parallel to the emergence of autonomous cars, autonomous trucks are also in development and are likely to revolutionize the transport sector and in particular freight transport. Again, Tesla wants to be a pioneer in this domain with the Tesla Semi, which is supposed to arrive around 2021 on the market. Autonomous vehicles are expected to improve road safety by drastically reducing the number of incidents. Furthermore, they are likely to be an excellent ally in the fight against climate change by improving the road transport efficiency and reducing greenhouse gas emissions.

The aim of this work is to emphasize the gains of using autonomous trucks over regular trucks for freight transport. These gains are mainly based on the prime costs of such trucks but also, in a lesser extent, to the greenhouse gas emissions. To do so, two similar trucks, one with a driver and an autonomous one, are compared for different delivery situations, from national to international level. While the autonomous truck can drive continuously, the driver is submitted to the European legislation on driving and working hours. Their respective journey, which visits a set of $n$ clients selected randomly, is computed by minimizing either the driving time or the driving distance. To each route, a monetary cost is associated which serves as a basis for the comparison between the two trucks.

# Contents

# Nomenclature

ADP Autonomous truck with driving Distance Minimization

API Application Programming Interface

ATP Non-Autonomous truck with driving Distance Minimization

CNR Comité National Routier

DP Dynamic Programming

NADP Autonomous truck with driving Time Minimization

NATP Non-Autonomous truck with driving Time Minimization

TSP Travelling Salesman Problem

VRP Vehicle Routing Problem

# 1 Introduction

## 1.1 Aim of this Master's thesis

The world is on the edge of an economical breakthrough with the emergence of autonomous cars. Such vehicles do not, or for a very short time, require the driver's attention or even not require a driver at all. This new era of vehicles is largely promoted by Tesla's eccentric CEO Elon Musk and the promotion of the S 3 X Y series of vehicles which include an autopilot mode. In parallel to the emergence of autonomous cars, autonomous trucks are also in development and are likely to revolutionize the transport sector and in particular freight transport. Again, Tesla wants to be a pioneer in this domain with the Tesla Semi, which is supposed to arrive around 2021 on the market. Autonomous vehicles are expected to improve road safety by drastically reducing the number of incidents. Furthermore, they are likely to be an excellent ally in the fight against climate change by improving the road transport efficiency and reducing greenhouse gas emissions.

The aim of this work is to emphasize the gains of using autonomous trucks over regular trucks for freight transport. These gains are mainly based on the prime costs of such trucks but also, in a lesser extent, to the greenhouse gas emissions. To do so, two similar trucks, one with a driver and an autonomous one, are compared for different delivery situations, from national to international level. While the autonomous truck can drive continuously, the driver is submitted to the European legislation on driving and working hours. Their respective journey, which visits a set of $n$ clients selected randomly, is computed by minimizing either the driving time or the driving distance. To each route, a monetary cost is associated which serves as a basis for the comparison between the two trucks. The report is constructed such that everything is set to answer the questions of the gains of using autonomous trucks for freight transport.

First, in the rest of this introduction, some examples of actual autonomous trucks are given and then a PESTEL analysis is made. Second, in Section 2, a state of the art is given for the linear and dynamic programming to solve the well known Travelling Salesman Problem as well as the extension to a fleet of vehicle: the Vehicle Routing Problem. This section also tackles the problem of introducing the European legislation on driving and working hours into these two optimization problems. Third, in Section 3, the modelling of the Vehicle Routing Problem for both autonomous and non autonomous trucks is made. In this section, three case scenarios, from national to international level, are detailed and all the roads between any two cities are computed using an API. Furthermore, all the costs associated to each truck are explained for the variable and fixed costs but also the salary costs. Then, $CO_2$ emissions are briefly discussed. Fourth, in Section 4, the results for each case scenario are displayed and discussed. Finally, conclusions are drawn on the potential gains of using autonomous trucks for freight transport.


Figure 1.1 – The Pod, during the partnership with Lidl to transform its supply chain to be emissions-free ©Einride, 2020

## 1.2 Examples of autonomous trucks

An autonomous truck is a truck which is equipped with a technology such that it is capable of driving partly or entirely by itself. After the emergence of autonomous cars, autonomous trucks should make an enormous breakthrough in freight transport. Besides the Tesla Semi, several examples are already under development and even operational.

For example, The Pod from the Einride start-up in Sweden, shown in Figure 1.1, is the first all-electric autonomous vehicle for transport that can operate on public roads since 2019 and whose information can be retrieved from https://www.einride.tech/pod/. This truck has a payload of 16 tons and a capacity of 15 to 18 pallets. An interesting aspect of The Pod is that its charging is automated and therefore no one is needed to plug it, and per charge, the truck can cover up to 180 km.

Another example is Vera from Volvo, presented in 2018, which is a cockpit-less tractor and whose primary mission is the transport of goods from a logistic centre to a port terminal. More details can be retrieved from https://www.volvotrucks.com/vera.html.

The last example discussed is the autonomous truck from plus.ai, shown in Figure 1.3, which is already in service in the U.S.[1]. They are producing level 4 fuel-powered autonomous

[1] https://plus.ai/index.html

Figure 1.2 – Vera, the autonomous and electric vehicle from ©Volvo, 2020

trucks, which means that they are fully autonomous, but the driver can still take the control at any moment and, therefore, a cockpit is still present. Furthermore, according to their website, up to 25% of fuel can be saved, decreasing the $CO_2$ emissions by the same amount. Such a truck has already been capable of crossing the U.S., 4500 km, in less than three days[2] while it would have taken 7 days for a driver driving at 70 km/h and 9 hours a day.

In the present work, the autonomous truck considered presents the same characteristics as the *plus.ai* autonomous truck, *i.e.* a truck that is similar to a regular heavy truck, with a cockpit and the same driving capabilities. Doing so, it avoids the comparison between an electric and a fuel powered truck, for which it can be tedious to compare their respective costs as well as the "fuel" price. Here, the emphasis is made only on the autonomous trait of the truck, which is therefore not submitted to driving hours regulations as it is the case for drivers.

## 1.3 PESTEL analysis of the autonomous truck

**Political factors:** At the European level, the political factors are mainly driven by the recent Green Deal that aims at reducing drastically the greenhouse gas emissions, which is detailed in the paragraph about the environmental factors. Autonomous trucks could be in adequacy with these environmental priorities and therefore, the political climate should be favourable.

[2] https://www.fleetowner.com/technology/autonomous-vehicles/article/21704522/plusai-completes-crosscountry-delivery-with-selfdriving-truck


Figure 1.3 – The plus.ai self driving truck ©plus.ai, 2020

In Belgium, the situation is different. The country is often subject to political crises, such as in 2010-2011 when it took 541 days to form a government. Again, at the moment, the country is still incapable of forming a government due to huge political discrepancies between the North and the South of Belgium. While in the North, the tendency is to vote for the right wing, in the South, this is the opposite with a domination of the left wing political parties. Therefore, obtaining an agreement for a subject such as autonomous trucks from a government can take a long time. Furthermore, the influence of the syndicates is very strong in Belgium and autonomous trucks would leave lots of truck drivers unemployed unless redirected. There is no doubt that the syndicates will go against measures allowing the replacement of regular trucks for autonomous trucks. Such a political climate could be a brake towards the emergence of autonomous trucks in Belgium, leaving the country behind others in this domain.

**Economic factors:** It is estimated that there is around 3 million truck drivers in Europe for approximately 6 million trucks[3]. Totally replacing these trucks by fully autonomous trucks would mean leaving these truck drivers unemployed unless they are redirected. This would not be the first drastic reduction of employment in a sector due to automation. For example, in the U.S., 5 million workers from the manufacturing domains lost their job after 2000[4]. 40% of these people did not find a new job and the same phenomena is likely to happen to the future unemployed truck drivers since they typically only have a high school diploma. Furthermore, the impact

[3] https://www.acea.be/statistics/article/vehicles-in-use-europe-2017

[4] https://evonomics.com/what-will-happen-to-truck-drivers-ask-factory-workers-andrew-yang/

on employment will depend on how quickly these new technologies will be introduced into the market. The slower they are introduced, the higher the chances are that the negative impacts will be absorbed by the economic system (Raposo et al., 2018).

However, it is expected that the market of autonomous vehicle will grow exponentially and many jobs should be created, requiring new skills. For example, there will be an increase in the Information and Communication Technologies domains. Furthermore, experienced drivers could be useful for the monitoring of autonomous trucks via remote control (Raposo et al., 2018).

In addition to that, several sectors should expect an important growth such as electronics and software with an increase of the production and the sales of new elements that will be needed for self-driving vehicles. Also, the telecommunication, data services and digital media can also expect a significant growth, especially with the introduction of the 5G network. This connectivity between cars and infrastructure could generate revenues up to $450 to $750 billion by 2030 according to MacKinsey&Company[5]. Finally, the freight transport should expect a drastic reduction of their costs with the introduction of autonomous trucks. In fact, the two most expensive elements of the prime cost of truck journey are the driver' salary and the fuel consumption, which will both be reduced.

However, two sectors should be negatively affected by the introduction of autonomous vehicles, namely the insurance sector and the maintenance and repair sector. In fact, we expect a drastic reduction of road incidents and therefore an improvement of the road safety, which will result in discounts in motor vehicle premium insurances and less crash-related repairs. These reductions could represent a total loss of 53 billion euros by 2050 for the insurance sector (Raposo et al., 2018).

Finally, a last economic aspect to discuss is the sharing of autonomous trucks. In fact, since no driver would be needed for autonomous trucks, they would not need to come back to the depot for the driver to come home. Instead, when arriving to its destination and when unloaded, it could be rent by another society and start another delivery. In this new economic model, some societies across Europe would only own such trucks and rent them to other societies which need to make deliveries, and the closest available truck would be selected. This model could drastically improve road transport efficiency.

**Social factors:** Ever since the first model of self-driving vehicles were developed, ethical questions arose, especially around the unavoidable crash. A famous example is the question of who to save between the driver and a pedestrian if a crash were to be unavoidable. In the last 5 years, a few crashes involving automated vehicles happened, mainly in the U.S. and with Tesla models. The vehicles involved were level 2 autonomous cars, which is a level in which

[5] https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/accelerating-the-car-data-monetization-journey

the driver has to be fully aware at all times during the drive. These ethical questions are directly linked to how the algorithms piloting the vehicles should be constructed and can be called *moral algorithms* (Ryan, 2020). Although these crashes are infrequent, a fear towards these vehicles grew into some people's mind, despite the fact that for 95% of road incidents involve human errors. Self-driving cars could help drastically reducing the number of deaths on the roads and improve their safety.

Concerning traffic Jam, autonomous trucks could help reducing them since they could be stopped at these peak times, privileging night driving when possible. Furthermore, fatigue is key parameter in truck related incidents, especially at night when fatigue is at the highest. This would no longer be an issue with fully autonomous trucks.

**Technological factors:** In the car industry, it is usual to talk about 6 levels of automation, which are described in Figure 1.4 and that can be summarized as follows:

- Level 0: only the driver controls the vehicle for both longitudinal and lateral controls, there is no automation;
- Level 1: both the driver and car share the control of the vehicle. For example, using the cruise control, the vehicle is able to maintain a constant speed;
- Level 2: the car has full control of itself but the driver must monitor the driving and be ready to intervene at any time;
- Level 3: the driver does not need to monitor the vehicle anymore and can do other activities during the drive. However, he may be still needed for specific actions;
- Level 4: the driver's attention is no longer required. He can sleep, for example, during the trip;
- Level 5: the car has full control of itself and no driver is required anymore.

In order for these levels to be reached, especially levels 3 to 5 which are the future of autonomous vehicles, existing and new technologies are involved. For example, the 5G network is being deployed worldwide which will ensure a rapid communication with hardly any latency for large amounts of data between cars and its environment, such as the other cars or the infrastructures.

Concerning the positioning of the vehicles, Galileo, which is Europe's Global Navigation Satellite System, will be a critical component for autonomous driving in Europe, according to some high-ranked representatives from the European Commission and from the automotive industry[6], thanks to a high precision service, of the order of 20 cm.

[6] https://www.gsa.europa.eu/newsroom/news/galileo-critical-autonomous-driving

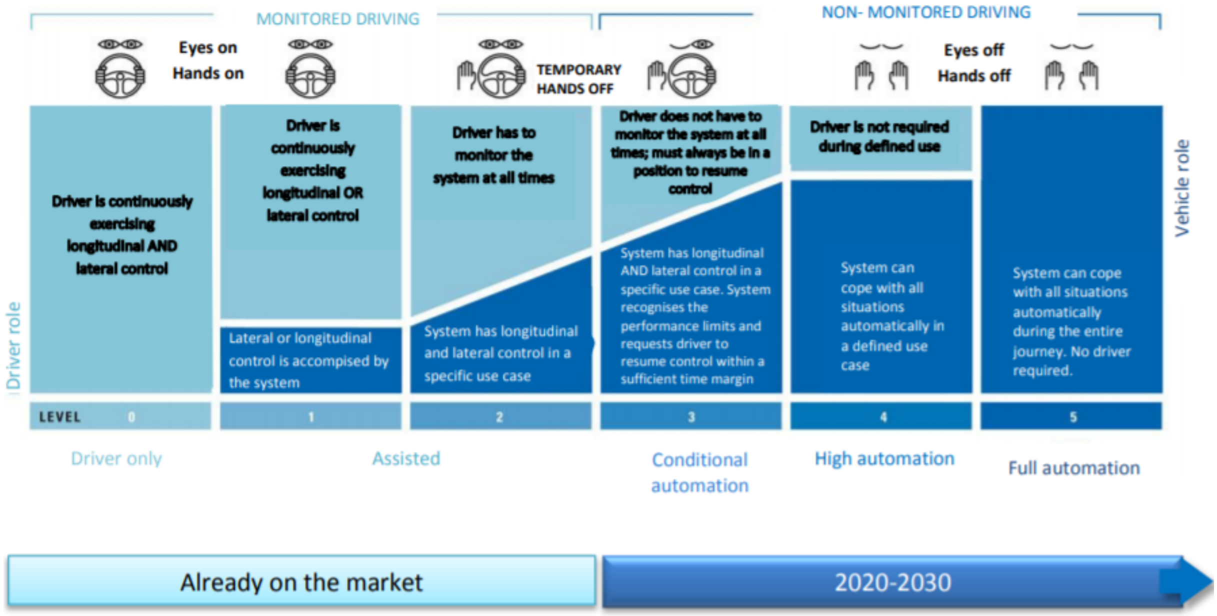



Figure 1.4 – Different levels of automation ©Society of Automotive Engineers

Nevertheless, the GPS itself is not sufficient and other on-boards systems are required such as cameras, radars and lidars in order to correctly monitor the surrounding environments and approximate the distance with respect to other cars and avoid all types of obstacles.

**Environmental factors:** On December $11^{th}$ 2019, the European Commission presented The European Green Deal. The goal of this Green Deal is to make Europe the first climate-neutral continent by 2050. To do so, the European Commission proposed in March 2020 a framework to achieve this climate neutrality, the so-called Climate Law (European Commission, 2020). In September 2020, they should present an impact assessed plan with a target of reducing greenhouse gas emissions by at least 50% compared with 1990 levels for 2030. Between 2030 and 2050, the Commission will take several measures in order to obtain the 2050 objective, which is threefold[7]:

1. *No net emissions of greenhouse gases*;
2. *The European economic growth is decoupled from resource use*;
3. *No person and no place is left behind.*

President Ursula von der Leyen said about this Green Deal: "*The European Green Deal is our new growth strategy – for a growth that gives back more than it takes away. It shows how to transform our way of living and working, of producing and consuming so that we live healthier and make our businesses innovative. We can all be involved in the transition and we can all benefit from the opportunities*". Autonomous trucks could really comply with this new growth

[7] https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en

strategy since the net distance covered for a same task compared with regular trucks would decrease as well as $CO_2$ and other greenhouse gas emissions. In fact, in 2014, road transport accounts for 19.5% of the greenhouse gas emissions, as it can be seen in the graph from Figure 1.5 which presents the official data from the European Environment Agency. Two priorities can be cited towards a reduction of these emissions[8]. First, by moving towards zero-emission vehicles, from their creation to their end of life. Furthermore, these vehicles should be powered by sustainable energies. At the moment, autonomous trucks tend to be fully electric, which is a first step into avoiding the use of fossil fuels, providing that the electricity consumed comes from sustainable energies. Second, we need to increase the efficiency of the transport system, which is totally doable with autonomous trucks. In fact, European regulations on driving and working hours for drivers make the actual road transport system quite ineffective.

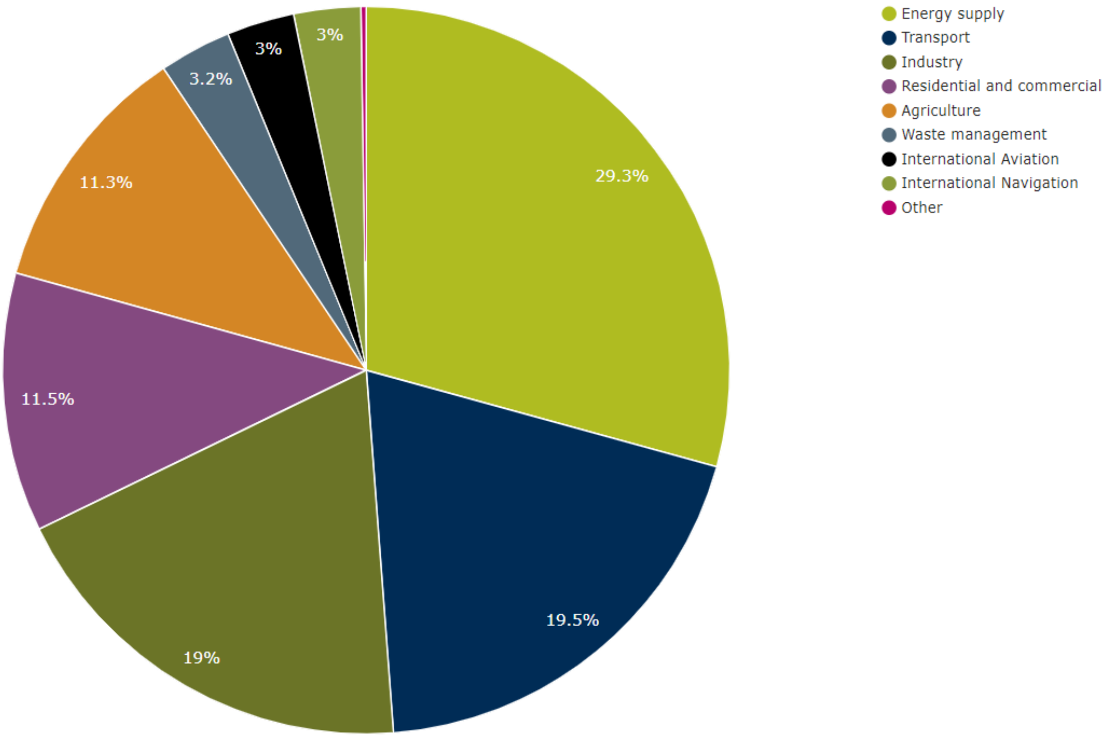



Figure 1.5 – Greenhouse emissions share per main sectors in 2014 ©European Environment Agency

**Legal factors:** At the moment, the Vienna Convention, which was concluded in 1968 and entered into force in 1977, is of application in several countries of the United Nations and especially in Europe. Article 8 of this convention states that *Every moving vehicle or combination of vehicles shall have a driver* which means that a level 5 self-driving vehicle is not legal yet (United Nations, 1968). However, as the technologies are evolving rapidly, the European Union is actively working on a common legislation and the European Commission has already written a communication on automated mobility (European Commission, 2018). Following that, Wimp

[8] https://ec.europa.eu/clima/policies/transport_en

van de Camp, from the Dutch EPP, wrote a report which has been adopted by the parliament in January 2019 (van de Camp, 2018). Some elements of this report are emphasized by the European Parliament on their website[9]:

- *EU policies and legislation concerning automated and connected transport should cover all transport modes, including short-sea shipping, inland waterway vessels, drones transporting goods and light rail systems;*
- *Standardisation efforts at international level need to be further coordinated to ensure safety and the interoperability of vehicles across borders*;
- *Event data recorders should be compulsory in automated vehicles to improve accident investigations and tackle the issue of liability*;
- *To increase the trust of Europeans in driverless vehicles, rules covering data protection and ethics in the automated transport sector should be developed without delay*;
- *Special attention should be given to the development of self-driving vehicles that are accessible for people with reduced mobility or disabilities*.

[9]https://www.europarl.europa.eu/news/en/headlines/economy/20190110STO23102/self-driving-cars-in-the-eu-from-science-fiction-to-reality

# 2 State of the art

## 2.1 Introduction

In this section, several aspects are discussed. First, since the autonomous truck is studied for several routes across Belgium or Europe, an optimized route must be selected among every possibility in order to minimize a specific cost function, such as time or distance. Therefore the Travelling Salesman Problem (TSP) is introduced. On the one hand, it is solved using a linear programming procedure, and on the other hand, using a dynamic programming approach, which offers much more flexibility in its implementation. Second, the TSP is extended to the Vehicle Routing Problem (VRP) in the case of a fleet of $m$ vehicles. Finally, the problem of the inclusion European regulations on driving and working hours in an optimization procedure is tackled.

## 2.2 Travelling salesman problem

The TSP is an optimisation problem, first considered in the 1930s, which has been largely studied in the literature. It is the very basis of any route planner software such as RouteXL, Driving Route Optimizer or GraphHopper. The problem is the following: finding the shortest route passing by $n$ points. Each point must be visited once and only once and the route must end at the starting point. For instance, in Figure 2.1, the optimal route, starting and ending at the depot, then passing by the remaining $n-1$ points is shown. Each point has a pair of coordinates, namely the latitude $\phi$ and the longitude $\psi$, from which it is possible to compute the distance as the crow flies between two points, $i$ and $j$, using the haversine formula:

$$d_{ij} = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_j - \phi_i}{2}\right) + \cos(\phi_i)\cos(\phi_j)\sin^2\left(\frac{\psi_j - \psi_i}{2}\right)}\right) \tag{2.1}$$

where $r$ is the Earth radius. The distance $d_{ij}$ is then linked to the time such that $t_{ij} = d_{ij}/v$, where $v$ is the speed of the salesman, and it is now possible to build a distance matrix, or equivalently a time matrix, whose size is $n \times n$. In this case, the matrices are symmetric, but it is easy to imagine a real-life scenario where going from point $i$ to point $j$ does not involve going over the exact same route at the exact same speed.

Most optimization problems can be divided into two categories:

1. Polynomially solvable problems: the time complexity for solving the problem is a polynomial of the input size;

2. NP-hard problems: there is no algorithm that has a polynomial time complexity.

Unfortunately, as simple as the statement of the problem is, the TSP was proven to be NP-hard by Karp (Karp, 1972), which means that there is no algorithm that can give an optimal solution in a polynomial time.

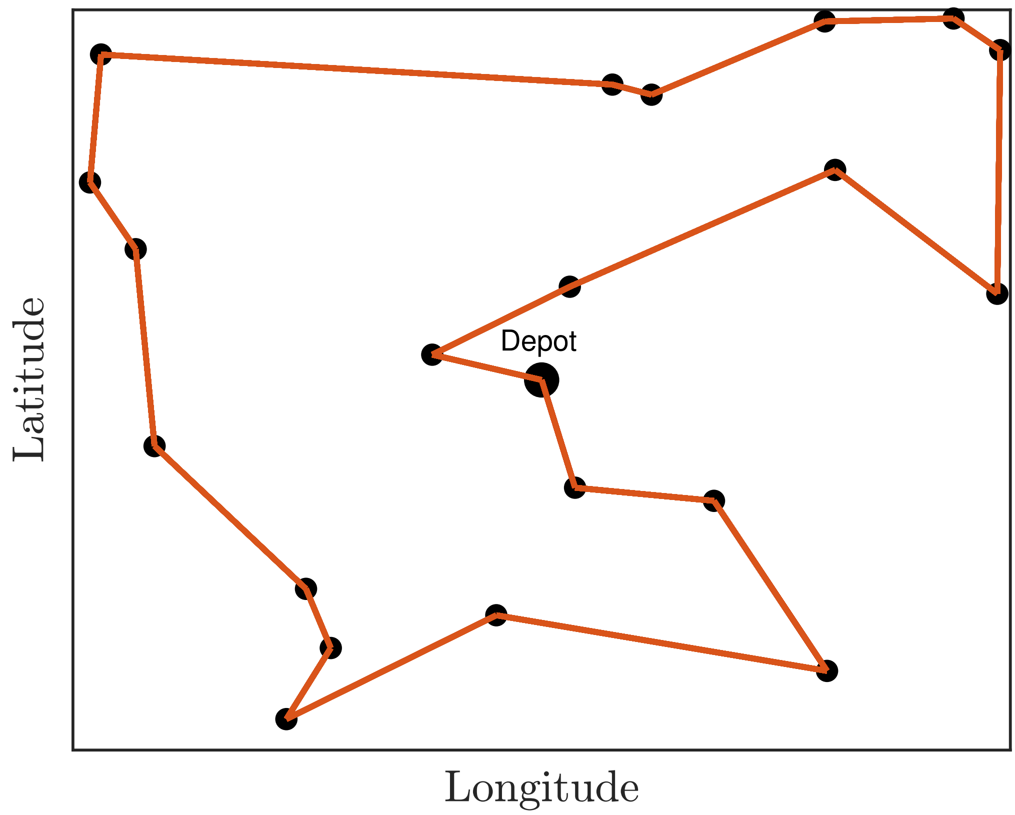



Figure 2.1 – Example of an optimal solution of the TSP for $n$ points

A naive approach to solve the problem would be to test every possibility separately. Such an approach is called the brute-force search and is not effective at all since the time complexity would be $\mathcal{O}((n-1)!)$, or $\mathcal{O}(((n-1)/2)!)$ if the distance matrix is symmetric. Fortunately, there exist several algorithms to solve this problem, gathered in two main classes:

1. Exact algorithms, which always give the optimal solution but with a not necessarily good time complexity;

2. Heuristic algorithms, which give a good feasible solution but does not guarantee the optimality.

In both cases, numerous algorithms exist for solving the TSP, and the problem can be rewritten using two different techniques described hereafter: linear programming and dynamic programming.

### 2.2.1 Linear programming

Several formulations of the problem exist and the one retained here is the Miller-Tucker-Zemlin formulation (Miller, Tucker, & Zemlin, 1960). Let $N$ be a set of $n$ nodes through which the route must pass and where the node 1 is the depot, *i.e.* the starting and ending node. We can define the variable $x_{ij}$ which is equal to 1 if the road from $i$ to $j$ is taken and 0 if not, where $i, j \in N$. All the cities are connected by an edge $e_{ij}$, forming a complete graph $E$. To each edge, we can attribute a cost $c_{ij}$, which can either be the distance $d_{ij}$ or the travel time $t_{ij}$. The ultimate goal of the TSP is to find the solution that minimizes the total cost function from Equation 2.2:

$$
\begin{aligned}
&\underset{x}{\text{minimize}} && \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij} && && (2.2)\\
&\text{subject to} && x_{ij} \in 0,1 && \forall i, j \in N && (2.3)\\
& && \sum_{i=1, i \neq j}^{n} x_{ij} = 1 && \forall i \in N && (2.4)\\
& && \sum_{j=1, j \neq i}^{n} x_{ij} = 1 && \forall j \in N && (2.5)\\
& && u_i - u_j + n x_{ij} \leq n - 1 && \forall i, j \in N, i \neq j && (2.6)\\
& && 1 \leq u_i \leq n && \forall i \in N && (2.7)\\
& && u_i \in \mathbb{Z} && \forall i \in N && (2.8)
\end{aligned}
$$

The first constraint, Equation 2.3 forces the variable $x_{ij}$ to be binary. The next two constraints, Equations 2.4 and 2.5 ensure that each node is only connected to two other nodes, namely the predecessor and the successor. Two other constraints called the subtour elimination constraints, Equations 2.6 and 2.7, ensure that the solution obtained is one tour joining all the nodes and not a set of subtours. To do so, additional dummy variables are added to the system: $\mathbf{u} = (u_1, \ldots, u_n)^T$, where $u_i$ represents the number of nodes visited when arriving at node $i$. Equation 2.8 makes sure that the variable $u_i$ is an integer.

Several exact or heuristic algorithms can be used to solve this optimization problem. Without going into the details, one can cite:

- cutting plane method;
- branch and bound;
- branch and cut;
- nearest neighbour search;
- genetic algorithms;
- simulated annealing.

Many of these algorithms are already implemented in lots of libraries or packaged for various programming languages such as *Matlab* and the *solve* package (which was used to compute the optimal solution in Figure 2.1), *Python* and the *SciPy* library, etc. However, libraries can be seen as black boxes where each condition of the problem must be written under the form of an equality or inequality which may not be always possible.

### 2.2.2 Dynamic programming

Dynamic programming (DP) was first introduced by Richard Bellman in 1954 (Bellman, 1954). The basic idea behind is to recursively decompose the problem into sub-problems, called states, and compute the optimal solution based on the optimal solutions of these sub-problems, which can also be decomposed into sub-problems until the the solution becomes trivial. These solutions are then computed and used to solve larger and larger states, until the complete problem. Before going any further, some notations as well as their definition must be brought. They are the ones from the book *Dynamic Programming for routing and Scheduling: Optimizing Sequences of Decisions* by Van Hoorn (Van Hoorn, 2016).

**Definition 2.1.** A sub-problem, or state, is defined by $\xi_\phi$. The subscript $\phi$ defines the specifics of the sub-problem.

**Definition 2.2.** Let $\zeta$ denote a solution. With $\zeta_\phi$ we define a solution to the sub-problem, or state, $\xi_\phi$. By $\check{\xi}_\phi$ we denote an optimal solution to the state $\xi_\phi$.

**Definition 2.3.** By $\zeta_{\rightsquigarrow i}$ we denote an expansion in the forward dynamic programming algorithm from solution $\zeta$ with $i$. This is a new solution of a larger sub-problem. The definition of $i$ depends on the specific problem.

In the case of a routing problem (TSP or VRP), it is important to define a set of nodes $N$, also called the customers, to be visited as well as a subset $S \subseteq N$. Here, the goal of the algorithm is to find the optimal solution $\zeta_N = \check{\xi}_N$ to the state $\xi_N$ which contains the sequence with all the nodes visited. To do so, as already explained, the whole problem is simplified into sub-problems, $\xi_S$, which admit an optimal solution $\zeta_S = \check{\xi}_S$.

In the case of the TSP, when the solution $\zeta_S = \check{\xi}_S$ of a certain sub-problem $\xi_S$ is found, it needs to be expanded to the next node $i \in N_{\setminus S}$. Therefore, the last visited node $l$ of the subset $S$ must be retained in order to be able to compute the cost $c_{li}$. To do so, $l$, is added in the subscript of the state as well as the current cost of the subset: $\xi_{S,l,c}$. Furthermore, the starting node $s$, is common to each possible road and thus, it is not included in the subset $S$. However, the ending node, also $s$, is included. The initial state to solve and expand is then $\xi_{\emptyset,s,0}$ and the final state, returning to the node $s$ is $\xi_{N,s,c_{total}}$.

To exemplify what has just been explained, let us assume a set of four nodes: $N = \{1,2,3,4\}$. There exists $2^n = 16$ possible subsets, where the first one is an empty subset and the last one where $S = N$. This can be put in a matrix form where the columns correspond to a subset $S$ and the lines correspond to the last visited node $l$. For each column, if the $i^{th}$ row is equal to 1, then the node $i$ is in the subset $S$. $M$ can be defined as:

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & \boxed{1} & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{2.9}$$

Several elements are emphasized here. First, the element $M_{2,6}$, in red, corresponds to the state $\xi_{\{1,2\},2,c}$ since for the $6^{th}$ column, the first and second rows are equal to 1 and the two other rows are equal to 0. Second, if the starting node is $s = 1$, then 1 cannot be in the subset $S$ except if $S = N$, otherwise it would mean that the route ends before visiting the remaining nodes. Therefore, all the states, whose last visited node is 1, emphasized in blue, are not feasible. On the contrary, since the route must finish in 1, the only feasible state of the last column ($S = N = \{1,2,3,4\}$) is for the $1^{st}$ row, in green. The other elements of the last column, in yellow, are not feasible.

Practically, for the implementation, we define a state variable matrix, based on the $M$ matrix, in which the current state solutions are stored and which is updated each time a new better solution for the current state is found or if an optimal solution is expanded to the next node. Since this is a forward DP algorithm, the matrix is updated from the left to the right, *i.e.* nodes are added to the route. First, as in the matrix $\Xi$ from equation 2.10, all the solutions are empty, the current node is $s = 1$ and no other node has been visited yet.

$$\Xi = \begin{pmatrix} \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \end{pmatrix} \tag{2.10}$$

The next step is to expand the empty solution to the next remaining nodes, either 2, 3, or 4. Since there is only one possibility per subset $S$, the solution is trivial and the cost for each state is $c_{1l}$. The solutions are stored as in Equation 2.11.

$$\Xi = \begin{pmatrix} \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \check{\xi}_{S,2,c} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \check{\xi}_{S,3,c} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \check{\xi}_{S,4,c} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \end{pmatrix} \tag{2.11}$$

Then, each state is expanded to a next node. For example, state $\xi_{S,2,c}$, whose subset $S$ is $\{2\}$, can be expanded to other subsets, either $S = \{2,3\}$ or $S = \{2,4\}$. If we refer to the matrix $M$ from Equation 2.9, these states correspond to the indexes $(3,9)$ and $(4,10)$ respectively. Note that it cannot be expanded to the indexes (2,9) or (2,10) since node 2 is no longer the last visited node. The cost for each state is still trivial since we only need to add the cost $c_{2,3}$ or $c_{2,4}$

respectively. This expansion is shown in Equation 2.12[10].

$$\Xi = \begin{pmatrix} \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \check{\xi}_{S,2,c} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \check{\xi}_{S,3,c} & \emptyset & \emptyset & \emptyset & \emptyset & \check{\xi}_{S,3,c} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \check{\xi}_{S,4,c} & \emptyset & \emptyset & \emptyset & \emptyset & \check{\xi}_{S,4,c} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \end{pmatrix} \tag{2.12}$$

This procedure is repeated continuously and each time multiple solutions exist for a state $\xi_{S,l,c}$, only the optimal one is retained in the matrix $\Xi$. Finally, the matrix is filled in until the optimal solution, still emphasized in green, is found as in Equation 2.13:

$$\Xi = \begin{pmatrix} \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \boxed{\check{\xi}_{N,1,c}} \\ \emptyset & \emptyset & \check{\xi} & \emptyset & \emptyset & \check{\xi} & \emptyset & \emptyset & \check{\xi} & \check{\xi} & \emptyset & \check{\xi} & \check{\xi} & \emptyset & \check{\xi} & \emptyset \\ \emptyset & \emptyset & \emptyset & \check{\xi} & \emptyset & \emptyset & \check{\xi} & \emptyset & \check{\xi} & \emptyset & \check{\xi} & \check{\xi} & \emptyset & \check{\xi} & \check{\xi} & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \check{\xi} & \emptyset & \emptyset & \check{\xi} & \emptyset & \check{\xi} & \check{\xi} & \emptyset & \check{\xi} & \check{\xi} & \check{\xi} & \emptyset \end{pmatrix} \tag{2.13}$$

Mathematically, this can be summarized in the recurrence relation of Held and Karp (Held & Karp, 1962) and Bellman (Bellman, 1962):

$$C(\check{\xi}_{S,i}) = \begin{cases} c_{si}, & \text{if } |S| = 1 \\ \min_{j \in S \setminus \{i\}} \{C(\check{\xi}_{S \setminus \{i\},j}) + c_{ji}\}, & \text{otherwise} \end{cases} \tag{2.14}$$

The time complexity of this algorithm is $\mathcal{O}(n^2 2^n)$, which is the best known time complexity. Equation 2.14 results in algorithm 2.1 developed in (Van Hoorn, 2016).

## 2.3 Vehicle routing problem

The VRP is the extension of the TSP in the case of a fleet of $m$ vehicles. Each vehicle $v_i \in V$ has an origin $o_i \in O$ and a destination $d_i \in D$, which are not necessarily the same. There is still $n$ nodes, or customers.

There exists several variants of the VRP, which all exhibit particularities and can be combined together. Among them, one can cite:

- Capacitated VRP: each customer needs a certain quantity $q_i$ to be delivered and each vehicle has a maximum capacity $Q_i$.
- Time Windows VRP: each customer requires to be delivered within a specific time window, *i.e.* its opening hours.

[10] For a better readability, the subset is always written $S$ but in reality each column correspond to a different subset, therefore, $\check{\xi}_{S,3,c}$ from indexes (3,4) and (3,9) are not the same. The first one corresponds to a route $1 \Rightarrow 3$, and the second one to a route $1 \Rightarrow 2 \Rightarrow 3$

**Algorithm 2.1** Forward DP algorithm for the TSP

**Input:** An instance of the TSP defined by a complete graph $G = (N, E)$ and a distance $c_{ij}$ for edges $e_{ij} \in E$
**Output:** A sequence $\zeta$ associated with an optimal route for the TSP
$\breve{\xi}_{\emptyset,s} = \zeta_{\emptyset,s,0}$
**for** $L = 0$ **to** $|N| - 1$ **do**
  **for all** $S \subset N$ **such that** $|S| = L$ **do**
    $S' = S$
    **if** $S' = \emptyset$ **then**
      $S' = \{s\}$
    **for all** $i \in S'$ **such that** $\breve{\xi}_{S,i} \neq \emptyset$ **do**
      **for all** $j \in N \setminus S$ **do**
        **if** $\neq s$ **or** $N \setminus S = \{s\}$ **then**
          **if** $\breve{\xi}_{S \cup \{j\},j} = \emptyset$ **or** $C(\breve{\xi}_{S,i}) + c_{ij} < C(\breve{\xi}_{S \cup \{j\},j})$ **then**
            $\breve{\xi}_{S \cup \{j\},j} = \breve{\xi}_{S,i} \rightsquigarrow j$
**return** $\breve{\xi}_{N,s}$

- Multiple compartment VRP: each customer needs a certain quantity of different types of product to be delivered;
- VRP with Pickup and Delivery: each customer requests is defined by a pickup point and a delivery point.

Solving the VRP is not much different than solving the TSP. The origins and destinations of each vehicle are added to the nodes to be visited which involves a larger set of nodes in the set $N = R \cup O \cup D$, where $R$ contains the nodes of the customer requests. In the VRP algorithm, one needs to make sure that each vehicle starts from its origin and ends at its destination. The origin of the next vehicle directly follows the destination of the previous one. The routes of each vehicle are then connected together to form the Giant-Tour Representation (GTR), which was introduced by Funke, Grünert and Irnish (Funke, Grünert, & Irnich, 2005). An example of a GTR solution of a VRP for a fleet of $m = 3$ vehicles and $n = 8$ clients is presented in Figure 2.2. As it can be seen, time complexity for the VRP increases compared to the TSP since we add $2m$ nodes, corresponding to the origin and destination of each vehicle. It is therefore $\mathscr{O}((n+2m)^2 2^{n+2m})$. Concerning the forward DP algorithm for the VRP, it is rather similar to the TSP algorithm presented in Algorithm 2.1 except that feasibility checks are added to ensure the origin and destination of each vehicle. This algorithm is displayed in Algorithm 2.2.

Using a DP algorithm, it is possible to not have a fixed cost function, such as in linear programming. It is really easy to change it according to the needs of the routing problem being solved. For example, a new variable can be added to the system, which retains the arrival time at each customer and permits to take traffic jam into account if, for the next node added, the vehicle drives during rush hour. The cost $c_{ij}$ can then be adjusted accordingly. It is also possible to add other constraints, such as the European regulations on driving, since a driver cannot drive

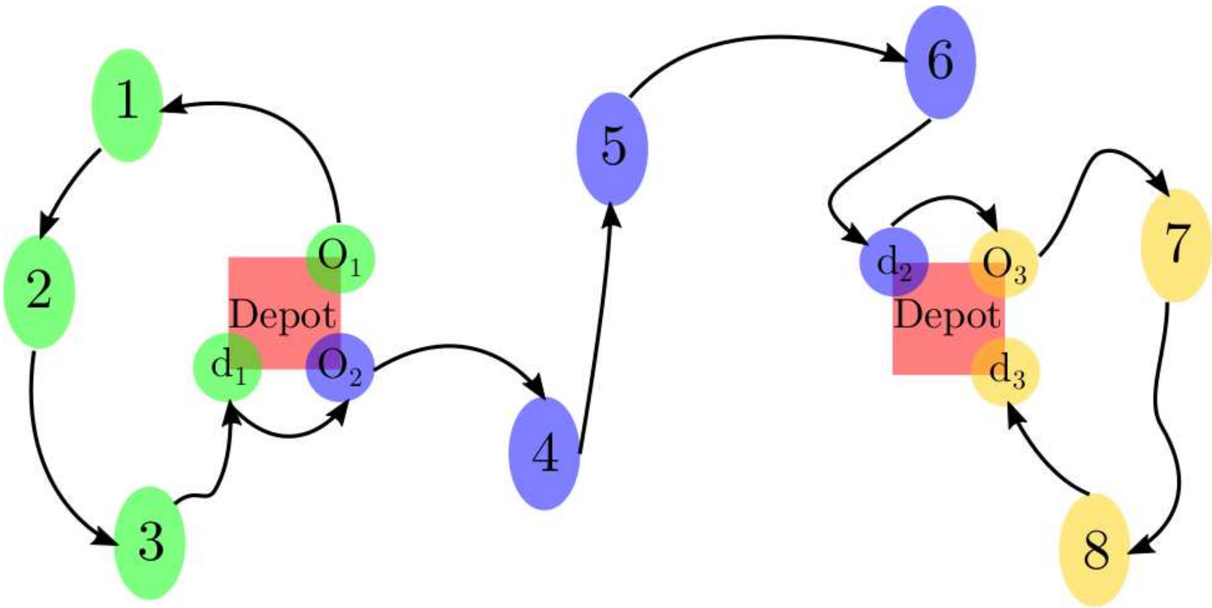



Figure 2.2 – Representation of a GTR solution of a VRP for a fleet of 3 vehicles and 8 customers

continuously but must schedule resting breaks. For these reasons, DP has been implemented in *Matlab* 2019 and used throughout the entire thesis to obtain the results.

## 2.4 European legislation on driving and working hours

A common legislation for all European countries has been established for the driving (Parliament & Council, n.d.-b) and working (Parliament & Council, n.d.-a) hours. The goal of this common regulation is threefold:

1. ensure a fair competition within the European Union;
2. improve road safety;
3. ensure drivers' good working conditions

The legislation on driving hours developed in the Regulation EC No 561/2006 and summarized on the official website of the European Union (Union, n.d.-a) and is detailed here:

- *Daily driving period shall not exceed 9 hours, with an exemption of twice a week when it can be extended to 10 hours;*
- *Total weekly driving time may not exceed 56 hours and the total fortnightly driving time may not exceed 90 hours;*
- *Daily rest period shall be at least 11 hours, with an exception of going down to 9 hours maximum three times a week;*
- *Daily rest can be split into 3 hours rest followed by 9 hour rest to make a total of 12 hours daily rest;*

**Algorithm 2.2** Forward DP algorithm for the VRP

**Input:** An instance of the VRP defined by a set of customer requests $R$ and a set of vehicles $V$ with for each vehicle $v_i \in V$ an origin $o_i \in O$ and a destination $d_i \in D$
A Graph $G = (N, E)$ with $N = R \cup O \cup D$ and a distance $c_{ij}$ for all edges $e_{ij} \in E$
**Output:** A sequence $\zeta$ associated which is the GTR an optimal solution for the VRP
$\check{\xi}_{\emptyset,d_m} = \zeta_{\emptyset,d_m,0}$
**for** $L = 0$**to** $|N| = 1$ **do**
  **for all** $S \subset N$ **such that** $|S| = L$ **do**
    $S' = S$
    **if** $S' = \emptyset$ **then**
      $S' = \{d_m\}$
    **for all** $i \in S'$ **such that** $\check{\xi}_{S,i} \neq \emptyset$ **do**
      **for all** $j \in N \setminus S$ **do**
        **if** $j = d_m$ **and** $N \setminus S \neq \{d_m\}$ **then**
          **continue**
        **if** $i \in D$ **xor** $j \in O$ **then**
          **continue**
        **if** $i = d_k \in D$ **and** $o_k \notin S$ **then**
          **continue**
        **if** $\check{\xi}_{S\cup\{j\},j} = \emptyset$ **or** $C(\check{\xi}_{S,i}) + c_{ij} < C(\check{\xi}_{S\cup\{j\},j})$ **then**
          $\check{\xi}_{S\cup\{j\},j} = \check{\xi}_{S,i} \rightsquigarrow j$
**return** $\check{\xi}_{N,d_m}$

- *Weekly rest is 45 continuous hours, which can be reduced every second week to 24 hours. Compensation arrangements apply for reduced weekly rest period. Weekly rest is to be taken after six days of working, except for coach drivers engaged in a single occasional service of international transport of passengers who may postpone their weekly rest period after 12 days in order to facilitate coach holidays;*

- *Breaks of at least 45 minutes (separable into 15 minutes followed by 30 minutes) should be taken after 4.5 hours at the latest.*

Concerning the working hours legislation, it is detailed in Directive 2002/15/EC and also summarized on the official website of the European Union (Union, n.d.-b):

- *Definitions of working time, periods of availability, place of work, mobile worker, self-employed driver, week, night time and night work;*

- *Maximum working week: 48 hours (this can be extended to 60 hours provided an average of 48 hours per week is not exceeded in any 4 month period);*

- *Breaks: not more than 6 hours should be worked consecutively without a break (at least 30 min when 6 to 9 hours are worked per day);*

- *Rest time: the provisions of Regulation (EC) 561/2006 are maintained;*

- *Night work: not more than 10 hours worked in any 24-hour period when a night shift is performed.*

As it can be seen, there is an exception to many rules. In the scope of implementing them into the Forward DP algorithm for the VRP, these exceptions may be cumbersome, although not impossible. In fact, several papers already discussed the implementation of these rules whether it be for linear programming (Kopfer & Meyer, 2009) or dynamic programming (Kok, Meyer, Kopfer, & Schutten, 2010) which was based on the work of Goel and Gruhn (Goel & Gruhn, 2005, 2006). In the context of this work, the paper from Kok (Kok et al., 2010) is used for the implementation of the European legislations for working and driving hours.

Kok et al. tried to stick as much as possible to the European legislation in their algorithm and it can be added to Algorithm 2.2 at the level of the feasibility checks since it adds new checks. For example, one needs to check whether the driver is able to go back to the depot before the end of his weekly working or driving hours after each client. Their algorithm, which covers a typical week of working and driving, is split into two parts. The first one, called *Basic break scheduling method* deals with the basic rules for both legislations on working and driving hours. The second part of the algorithm is called *Extended break scheduling method: outline changes with respect to the first algorithm* and deals with all the modified rules and exceptions of the legislation. In the present work, the focus is made on the first part of the algorithm, without taking all the exceptions into account, for simplicity reasons. The set of basic rules is summarized hereafter:

- The maximum daily driving time is 9 hours, split into two equal parts, between which a break of 45 minutes must be observed;
- The daily rest period should last at least 11 hours;
- The maximum weekly driving time should not exceed 56 hours;
- The maximum daily working period before a break of 30 minutes is observed is 6 hours. If the daily working period exceeds 9 hours, the break should last at least 45 minutes.
- The maximum weekly working time should not exceed 60 hours;

**Remark:** in their paper, Kok et al. only talk about a maximum weekly working time of 60 hours. In reality, it can last up to 60 hours provided that the average weekly working time does not exceed 48 hours over any 4 month period. Ergo, their algorithm does not seem to reflect exactly a typical week, since they consider the particular case when the driver works 12 hours more than on average.

Their algorithm *Basic break scheduling method* is detailed in Algorithm 2.3. It requires 6 new state variables:

- $t_{nbw}$: state variable that retains the accumulated working time before a break of 45 minutes is observed;
- $t_{nbd}$: state variable that retains the accumulated driving time before a break of 45 minutes is observed;
- $t_{nr}$: state variable that retains the accumulated time before a rest of at least 11 hours is observed;
- $t_{dd}$: state variable that retains the daily driving time;
- $t_{ww}$: state variable that retains the weekly working time;
- $t_{wd}$: state variable that retains the weekly driving time;

In addition to that, further information is needed. First, a service time $s_i$ for customer $i$ is defined. It takes any working time that has to be completed by the driver into account: loading and unloading the truck, cleaning the truck, etc. They are considered working time and known a priori. Second, the available time windows for each customer must be defined. Here, $e_i$ and $l_i$ are the opening and closing hours for customer $i$. Using all these variables and parameters and based on the European legislation, it is now possible to compute the final completion time $ct_i$ for each customer, *i.e.* the time at which the driver can leave from a customer and go to the next one or back to the depot.

In summary, Kok's algorithm basically computes the completion time for each customer added on the optimized route, based on the European legislation. Furthermore, it already includes the Time Window variant of the VRP and therefore, it does not have to be implemented anymore. Finally, completion time could be used to compute the next time travel, which would not be fixed a priori, but updated based on the departure time and the average traffic jam at that moment.

Concerning autonomous trucks, they would obviously not be submitted to the European legislation since they would be assisted by a vehicle operator working directly from the depot or another office performing shift or daily work, depending on when the autonomous trucks would be used.

## 2.5 Conclusion

It was shown in this section how to compute the TSP and the VRP using dynamic programming, which offers the best computational complexity for this kind of problems. DP is also very flexible in its implementation and the European legislation on driving and working hours can be included. Using theses algorithms, it is now possible to compute the optimized routes for both autonomous and non autonomous trucks by minimizing either driving time or driving distance.

**Algorithm 2.3** Basic break scheduling method

**Input:** State dimensions $t_{nbw}$, $t_{nbw}$, $t_{nbw}$, $t_{nbw}$, $t_{nbw}$, $t_{nbw}$, driving time $t_{ij}$ and service completion time $ct_i$ at customer $i$
**Output:** State dimensions $t_{nbw}$, $t_{nbw}$, $t_{nbw}$, $t_{nbw}$, $t_{nbw}$, $t_{nbw}$ and service completion time $ct_j$ at customer $j$

```
a_j ⇐ ct_i {Initialize arrival time}
δ_ij ⇐ t_ij {Initialize remaining driving time}
Δ = min(δ_ij, 6 − t_nbw, 4.5 − t_nbd, 13 − t_nr, 9 − t_dd) {Initialize remaining time before next break or rest}
ct_j ⇐ ∞{Initialize completion time}
if d_ij + d_j0 > 56 − t_wd then
   STOP {Adding customer j to the partial route is not feasible}
if d_ij + s_j + d_j0 > 60 − t_ww then
   STOP
while δij > Δ do
   if Δ = 9 − t_dd or Δ + 0.75 ≥ 13 − t_nr then
      a_j ⇐ a_j + Δ + 11
      t_dd ⇐ 0
      t_nr ⇐ 0
   else
      a_j ⇐ a_j + Δ + 0.75
      t_dd ⇐ t_dd + Δ
      t_nr ⇐ t_nr + Δ + 0.75
   t_nbw ⇐ 0
   t_nbd ⇐ 0
   δ_ij ⇐ δ_ij − Δ
   Δ ⇐ min(δ_ij, 6 − t_nbw, 4.5 − t_nbd, 13 − t_nr, 9 − t_dd)
if 13 − t_nr < max(0, e_j − a_j) + s_j or (6 − t_nbw < s_j and 13 − t_nr < max(0, e_j − (a_j + 0.75)) + s_j)
or a_j + 11 ≤ e_j then
   t_dd ⇐ 0
   t_nr ⇐ 0
   t_nbw ⇐ 0
   t_nbd ⇐ 0
   a_j ⇐ max(a_j + 11, e_j)
if 6 − t_nbw < s_j or a_j + 0.75 ≤ c_j then
   t_nr ⇐ t_nr + max(0.75, e_j − a_j)
   t_nbd ⇐ 0
   t_nbw ⇐ 0
   a_j ⇐ max(a_j + 0.75, e_j)
if a_j > l_j then
   STOP
```

# 3 Modelling the VRP for (non-)autonomous trucks

## 3.1 Introduction

The VRP using dynamic programming and including the European legislation on driving and working hours was implemented in Section 2. First, this section defines in which situation the trucks will evolve, for national and international transport. Second, a brief discussion about the $CO_2$ is made. Third, the actual routes joining any two cities are computed using an API. This API makes it possible to compute either the shortest or the fastest route. Finally, all the costs of each truck are detailed, which includes the variable and fixed costs but also the salary costs.

## 3.2 Case scenarios studied

In order to compare non-autonomous and autonomous trucks' optimized routes and their respective cost, three case scenarios are considered, to which a geographic area is attributed:

1. Belgium;
2. Belgium and close borders;
3. Europe.

Each case scenario can correspond to a certain real life application, from the national to the international level where a company, whose depot is located in Brussels, delivers its product to customers in Belgium and Europe. Each scenario is geographically different, except for the depot which is always in Brussels, which makes it possible to state in which case the autonomous truck presents the best alternative to the non-autonomous truck. For each scenario, a large number of cities is considered (around 40), which are presented in Table 3.1 where each city is labelled with a specific number. The longitude and latitude of each city can be retrieved from Tables A.1,A.2 and A.3. From these cities, only $n$ customers are selected randomly and numerous simulations are performed to get a good order of magnitude of the working and driving hours, the total cost as well as the number of vehicles needed to deliver these $n$ customers. Furthermore, this number $n$ can vary and, possibly, an optimal number of customers per delivery can be deduced. Finally, for each case scenario, different working hours and time windows can be imagined. These scenarios are described hereafter.

**Case scenario 1:** For the first scenario, in Belgium, the customers are only available from 8am to 5pm and the truck drivers can only work 9 hours per day and then must go back to the depot and then go home. If the delivery is not feasible in one day with one truck, new trucks are added. There are two ways to compute the total completion time. The first possibility is that there is only one driver who may take several days to complete the delivery. The second possibility is that the delivery is done in one day but it requires that the company making the delivery possesses enough trucks and available drivers.

**Case scenario 2:** The second scenario deals with customers located in Belgium, the Netherlands, Luxembourg and close to the borders with Germany and France. They are now available from 06am to 10pm, making two working shifts of 8 hours, from 6am to 2pm and from 2pm to 10pm. Here, the drivers can sleep on the roads but gets an indemnity for each night spent away. The deliveries can now take one week, from Monday 6am to Saturday 10pm, and the drivers can now complete maximum 56 driving hours or 60 working hours per week. Furthermore, the delivery is expected by the customer to be made within one week, so if one (non-)autonomous truck is not enough for a one week delivery, a new one is added.

**Case scenario 3:** The last scenario imagined takes place across continental Europe. Here, customers take the form of transport hub open 24 hours a day. This is the scenario that should reveal all the potential of autonomous trucks since they could drive continuously between two customers. Again, the delivery is expected to be made within one week and drivers must follow European legislation on working and driving hours.

Finally, some comments must be made regarding these scenarios. First, it is assumed that the service time $s_i$ for each customer lasts 30 minutes, no matter the scenario studied. This service is completed by the driver, in the case of a non-autonomous truck, and is considered a working time. In the case of an autonomous truck, a storekeeper from the customer's place is expected to do it. Second, it is assumed that each truck has enough capacity to cover the demand for the full delivery. The focus of this work was made on including the European legislation into the routing optimization rather than on capacity, which would increase the time complexity of the problem.

## 3.3 Fuel consumption of a truck

In 2019, 97.9% of new medium and heavy trucks used Diesel fuel (ACEA, 2020). Even though this kind motor will most probably change in the future with the rise of electric cars or the emergence of new types of motor such as the hydrogen motor, Diesel fuel is the type of fuel used in the present work for both autonomous and non autonomous trucks. The total fuel consumption of any mean of transport, from a point $A$ to a point $B$ can be computed as the integral over the route $|AB|$,as in Equation 3.1:

$$\int_A^B f_c(v(s))\mathrm{d}s \tag{3.1}$$

where $f_c(v)$ is the fuel consumption as a function of the speed $v$ and $s$ is a point of the route between $A$ and $B$. Unfortunately, it is very hard to get $f_c(v)$ since there is hardly any

| N° | Belgium | Belgium & Borders | Europe |
|---|---|---|---|
| 1 | Brussels | Bruxelles | Bruxelles |
| 2 | Aarschot | Liege | Ljubljana |
| 3 | Ostend | Mons | Zagreb |
| 4 | Waremme | Namur | Amsterdam |
| 5 | Wavre | Charleroi | Eindhoven |
| 6 | Louvain-La-Neuve | Arlon | Luxembourg |
| 7 | Arlon | Anvers | Paris |
| 8 | Nivelles | Bruges | Montpellier |
| 9 | Mons | Hasselt | Lille |
| 10 | Charleroi | Aarschot | Brest |
| 11 | Namur | Genk | Madrid |
| 12 | Yvoir | Maastricht | Barcelone |
| 13 | Ciney | Groningue | Seville |
| 14 | Huy | Amsterdam | Valence |
| 15 | Rochefort | La Haye | Rome |
| 16 | Val-Dieu | Utrecht | Milan |
| 17 | Achouffe | Arnhem | Naples |
| 18 | Orval | Rotterdam | Porto |
| 19 | Hasselt | Leeuwarden | Lisbonne |
| 20 | Chimay | Alkmaar | Berlin |
| 21 | Hal | Eindhoven | Munich |
| 22 | Buggenhout | Luxembourg | Hambourg |
| 23 | Bruges | Clervaux | Cologne |
| 24 | Anvers | Saint-Denis | Zurich |
| 25 | Sourbrodt | Rouen | Berne |
| 26 | Gouvy | Amiens | Vienne |
| 27 | Westvleteren | Reims | Graz |
| 28 | Achel | Lille | Prague |
| 29 | Hoegaarden | Calais | Brno |
| 30 | Melle | Metz | Varsovie |
| 31 | Audenarde | Saint-Quentin | Cracovie |
| 32 | Vichte | Sedan | Gdansk |
| 33 | Malle | Dunkerke | Budapest |
| 34 | Alken | Dortmund | Bratislava |
| 35 | Turnhout | Aachen | Kosice |
| 36 | Mol | Cologne | |
| 37 | Bouillon | Dusseldorf | |
| 38 | Saint-Hubert | Bonn | |
| 39 | Spa | Monchengladbach | |
| 40 | Genk | Duren | |
| 41 | | Duisbourg | |
| 42 | | Wuppertal | |
| 43 | | Heinsberg | |

Table 3.1 – Cities visited for the three case scenarios

documentation on the fuel consumption as a function of the truck speed available. Therefore, it is only possible to rely on an average consumption. It is commonly accepted that for heavy trucks, it oscillates around 32l/100km, or 0.32l/km (ICCT, 2018). This average consumption is used throughout this entire work even though the average speed is not always the same and the total consumption, $F_c$, is computed as in Equation 3.2:

$$F_c = 0.32 * Distance \tag{3.2}$$

This consumption represents high $CO_2$ emissions. In fact, if we consider the following simple complete combustion equation of a Diesel fuel:

$$4C_{12}H_{23} + 71O_2 \longrightarrow 48CO_2 + 46H_2O$$

It can be computed that for one litre of fuel, or 0.835kg, 2.64kg of $CO_2$ are exhausted. Alternatively, this represents 843g/km of $CO_2$ which is approximately 6 times more than a traditional car. In fact, trucks represents around 30% of the total road transport $CO_2$ emissions and are expected to go up to 40% by 2030 (Transport & Environment, 2015). This is huge and therefore, it is crucial to diminish these emissions in the fight against global warming. Minimizing the driving distance instead of the driving time would help, although, depending on the driver's or the vehicle operator's salary, it would be more expensive.

## 3.4 Time and distance by road transport using an API

As already explained in section 2.2, it is possible to compute the distance as the crow flies between two cities using the haversine formula from Equation 2.1. The only parameters needed are the longitude and latitude of each city. Then, the distance matrix can be assembled, as well as the time matrix by dividing the distance by the speed of the truck. Time and distance are thus proportional and the minimization of Equation 2.2 leads to the same results whether the cost parameter **c** is the distance or the time. In reality, though, this is not the case. Many parameters come into play: traffic jam, speed limits, closeness of the destination to a highway, etc. In order to be as realistic as possible, these distance and time matrices used in the rest of the report are the real ones, *i.e.* using road transport, and were obtained using an Application Programming Interface (API).

An API is an interface that specifies how two software, applications, websites can interact and exchange data between each other. In the case of a routing problem, many websites exist and are daily used by lots of consumers, such as *Google Maps*, *Waze*, *Mappy*, etc. And many of them propose an API that can be integrated in other applications, websites and so on. Unfortunately, in the case of a routing API, it is generally not free and the cost can go up to a few thousands Euros per year. For example, the *Mappy API* costs at leat 200€ per month (Mappy, 2020).

However, this API could have been extremely useful in the context of this work since it is possible to get much more information than just the distance, the time and the set of nodes through which the vehicle passes to join two cities. For instance, it is possible to define the type of vehicle, the type of fuel, the average consumption and the price of the fuel. Then it computes the fuel consumption and the corresponding price. The road is adapted to the kind of vehicle used and one can choose the fastest or the shortest road. Finally, it computes the total fee due in the case of a toll for every country. This last parameter is very hard to compute only based on the set of nodes through which the truck passes. *Google Maps* seems to be less expensive as it offers 200$ of use every month, then the price is proportional to the API calls consumed (Google, 2020). Its main advantage is that a departure time can be defined and the usual traffic jam is taken into account for the arrival time. However, the default vehicle is the car and no truck can be defined. Also, it is not possible to decide whether the output road is the shortest or the fastest.

The final choice for the API is the one proposed by the open-source routing library *GraphHopper* (GraphHopper, 2020). Several reasons for this choice can be cited. First, contrary to *Google Maps*, it is possible to request the fastest or the shortest route, and even a compromise between shortness and fastness. Second, as for *Mappy*, it is possible to define the type of vehicle used and the routes are computed accordingly. Finally, the first two weeks of use are free (with a limited daily use, but largely enough) which made it possible to compute thousands of routes, shortest and fastest routes, and store them into *.mat* files, which are *Matlab* files in which data can be stored.

The use of a *GraphHopper* API is fairly easy as it only consists in reading a URL in a programming language. The choice of the language is up to the user or depends on where the API is integrated. Here, it is read thanks to the *webread* function from *Matlab*. Once it is done, *GraphHopper* sends back a JSON formatted text file which contains all the data requested, such as the time, the distance and all the nodes visited. Each parameter of the request is directly written within the URL which is structured as:

https://graphhopper.com/api/1/[API TYPE]?[PARAMETERS]?key=[API KEY]

Three key words are highlighted in blue in this URL. The first one, [API TYPE], is the kind of API that needs to be called. In fact, *GraphHopper* proposes several API's, which all require different input parameters, to be specified at the location of [PARAMETERS] and three of them are interesting for the present work and are described hereafter. Finally, the API key, [PARAMETERS], is an authentication key attributed to each user which allows him to use the API. It is also used to calculate the number of API calls done by the user in order to not exceed his monthly or daily quota.

**Routing API:** this API is the most frequently used and is simply a navigation API that computes the best route between point $A$ and point $B$, as a GPS would work, and it is the API used throughout this report. The required input parameters are the latitude and longitude of each city that should be visited. Additional parameters can then be added. For example:

- *vehicle* for the type of vehicle used (car, bike, truck, etc.);
- *snap_prevention* if any road environment must be prevented such as bridges or tunnels. In this case, ferries were avoided;
- *weighting* to determine the best road to use, such as the shortest or the fastest;
- *instructions* to decide whether the driving instructions should be computed or not;

A URL for computing a route between Achouffe and Ostend in Belgium would look like this:

https://graphhopper.com/api/1/route?point=50.150781,5.746014&point=51.217579,2.913011
&vehicle=car&calc_points=true&points_encoded=false&weighting=fastest&key=XXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

where one can see the latitudes and longitudes for Achouffe and Ostend, the vehicle used (a car), all the nodes along the roads must be computed and the fastest route is sought. The API key is hidden here. The output of the request is a JSON formatted text that gathers all the information. In this example, the JSON output text is presented in Figure 3.1. Here, the number of visited nodes is 468, the total distance is 279820.217 $m$ and the total driving time is 9449115 $ms$. For readability reasons, some other information is hidden, such as the coordinates of the 468 nodes. This route is shown in Figure 3.2.

If a set of $n$ cities is considered, it is possible to do the same for every route, using a $for$ loop construct the $n \times n$ time and distance matrices.

**Matrix API:** this API could also be very useful in the context of solving the VRP. Here, instead of computing the road between two locations and sending back all its nodes, it takes as input a set of $n$ cities and directly computes the $n \times n$ distance and time matrices. It was not primarily used here for two reasons. First, it does not indicate which "best" route was calculated, *e.g.* the shortest, the fastest or a mix. Second, for a better visualization of the roads taken by the truck, the real routes are shown as in Figure 3.2. However, the Routing API does not always work and sometimes returns an *internal server error*, for which nothing can be done on the user's side. Therefore, when such an error occured, the time and distance between two cities were computed using the Matrix API and it is assumed that the shortest and the fastest routes are the same. In such a case, if the road has to be displayed, it is shown as a straight line.

```
{
    "hints": {
        "visited_nodes.sum": 468,
    },
    "info": {
        "copyrights": [
            "GraphHopper",
            "OpenStreetMap contributors"
        ],
        "took": 10
    },
    "paths": [
        {
            "distance": 279820.217,
            "weight": 13646.696189,
            "time": 9449115,
            "points_encoded": false,
            "points": {
                "type": "LineString",
                "coordinates": [
                    // Coordinates of the nodes of the route
                ]
            },
            "instructions": [
                // Detailed instructions
            ],
        }
    ]
}
```


Figure 3.1 – Example of a JSON formatted output text received after the call of an API for the routing from Achouffe to Ostend

**Route optimization API:** this API is used to solve the TSP or the VRP. Contrary to the two previous types of API, the Route optimization API does not only require to read a URL in which every parameter of the problem is inserted. Here, a JSON file has to be constructed with all kind of information such as:

- The vehicles: origin, destination, the type of vehicle, capacity, etc.
- The cities: latitude, longitude, name, time windows, demand;
- Cost matrix: the type of minimization that has to be performed, *e.g.* time or distance.

This JSON file has to be posted in the URL:

https://graphhopper.com/api/1/vrp?key=[API KEY]

This API is not used for the present work as it does not allow to include the European legislations on driving and working hours.

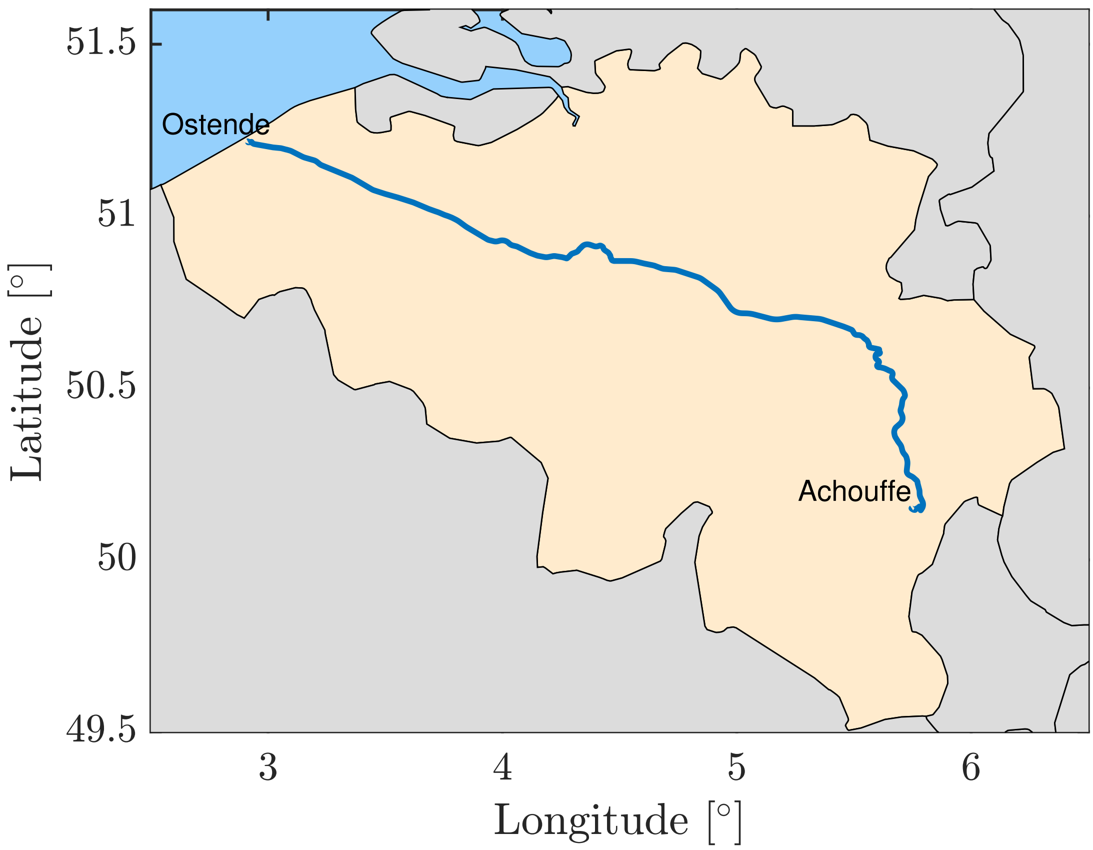



Figure 3.2 – Fastest route from Achouffe to Ostend using the Routing API from *GraphHopper*

## 3.5 Cost functions: time & distance minimization

Several cost matrices exist, as already explained, such as the time and distance matrices. Depending on how they are computed, *e.g.* using an API or with the haversine formula, the output sequence for the TSP or the VRP can be different. Let us take the example of one vehicle starting and ending from a depot in Brussels and visiting 9 customers. These customers are chosen randomly from the list in Table 3.1 for the first case scenario, *i.e.* taking place in Belgium. The resulting customers are: 7, 9, 10, 11, 22, 28, 29, 30 and 38. The classical TSP is solved, *i.e.* without any legislation on working and driving hours, time windows are service time. Three cost function are considered:

- minimum time by road transport;
- minimum distance by road transport;
- minimum distance as the crow flies.

The three output sequences are displayed in Figure 3.3 and the total time, total distance and the ordered output sequence are summarized in Table 3.2. First, it can be observed, by comparing Figures 3.3a and 3.3b, that the output sequence can be different whether driving time or distance is minimized. This difference in the output sequence has an impact on the total cost of the delivery. In fact, as it can be seen in Table 3.2, there is almost two hours of difference between the two situations and around 70 km of difference. It is up to the driving company to decide whether they want to save money on fuel consumption or on salary. The latter option is generally preferable to save money but, if the average fuel consumption is the same, the first option could help reduce the emission of greenhouse gases. The assumption of a constant average fuel consumption is important since for a shorter distance, small roads might be preferred to highways, and thus the average speed lowers which may lead to a suboptimal fuel consumption. For instance, in this case, the average speed would drop from 62 to 49 km/h.

Second, by comparing Figures 3.3b and 3.3c, the output sequence of the visited cities is the same. This is not very surprising since in Figure 3.3b, the shortest distance by land transport is computed, which is a straight line (rounded by the Earth curvature) and in Figure 3.3c, the shortest distance between two cities is sought. Even though this is not always the case, it is normal that these two situations give the same output. However, the total distance is 688.2 km and 579.36 km respectively, which represents a relative difference of 15.8%, which is not negligible. Furthermore, the total time is only 7h43 which is due to an assumption of an average speed of 75 km/h. If a more realistic speed of 50 km/h was to be used, the total time would have been 11h49 which also underestimates the real time by land transport. Finally, using the distance as the crow flies does not allow to distinguish a situation where the minimum time is sought and a situation where the minimum distance is sought, since in this case, they are proportional and would lead to the same answer.

| **Minimization** | **Time by road transport** | **Distance by road transport** | **Distance as the crow flies** |
|---|---|---|---|
| Total time | 12h16 | 14h03 | 07h43 |
| Total distance | 760.54 km | 688.2 km | 579.36 km |
| Sequence | $1 \to 30 \to 22 \to 28 \to 29 \to 7 \to 38 \to 11 \to 10 \to 9 \to 1$ | $1 \to 22 \to 30 \to 9 \to 10 \to 11 \to 38 \to 7 \to 28 \to 29 \to 1$ | $1 \to 22 \to 30 \to 9 \to 10 \to 11 \to 38 \to 7 \to 28 \to 29 \to 1$ |

Table 3.2 – Final results for different cost minimizations

Obviously, the error committed when using distance as the crow flies is not always 15.8%. In fact, it is possible to compute the mean and median errors as a function of the number of clients. This is done by repeating the simulations 100 times and for clients varying from 1 to 10 and chosen randomly each time. The results are displayed in Figure 3.4. For each number of client, the median is represented by a red straight line while the $25^{th}$ and $75^{th}$ percentiles are delimited by the blue box. The most extreme data points are represented by the whiskers and the outliers are represented by red crosses '+'. Finally, the mean is plotted using a star '*'.

First, for Figure 3.4a, for the first case scenario in Belgium, it can be observed that the median gradually converges around a relative difference of 17.5% and the boxes also gradually

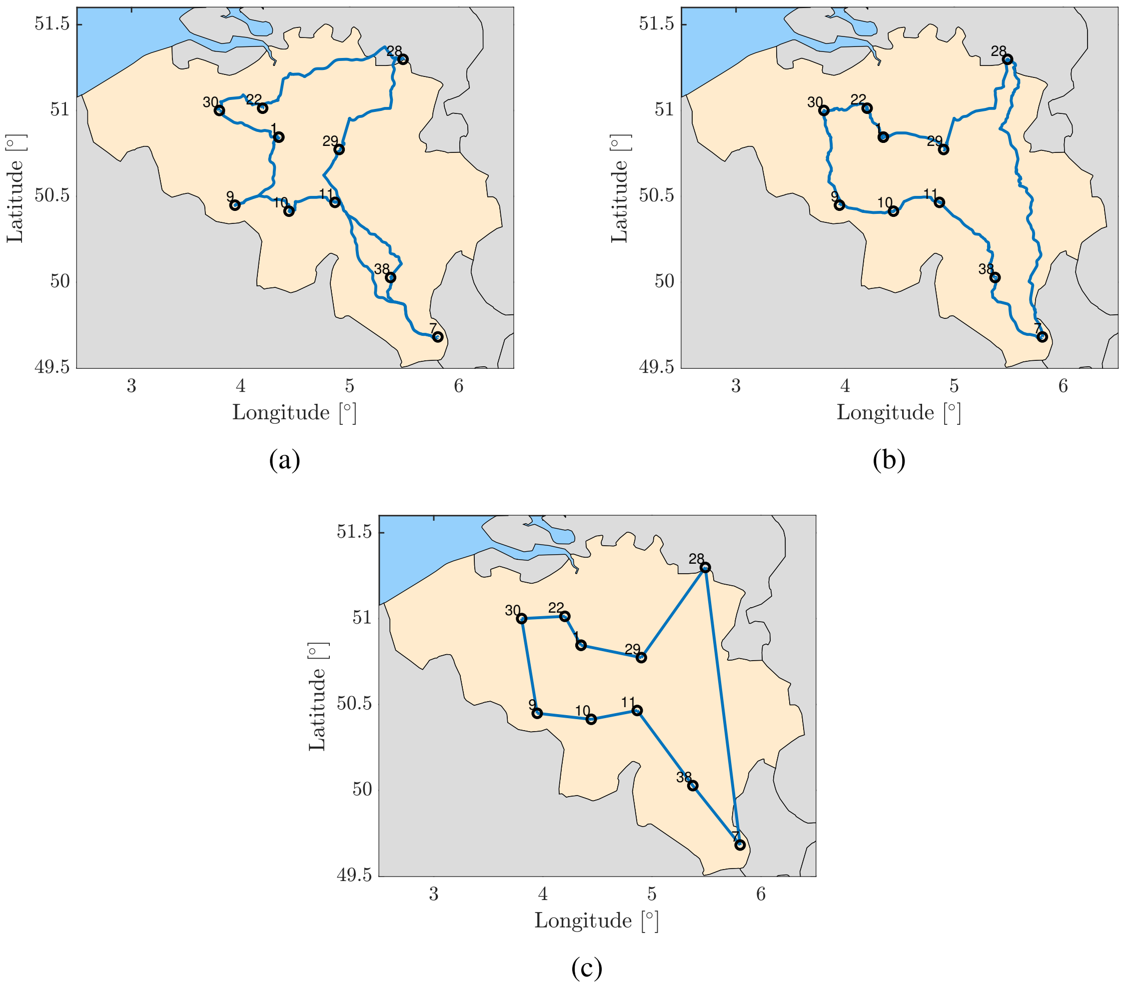



Figure 3.3 – Output sequence for three cost functions: (a) minimum time by road transport, (b) minimum distance by road transport, (c) minimum distance as the crow flies

shrink showing that the relative error is contained in a smaller and smaller range, which explains why the mean coincides well with the median. However, it still varies from 15 to 20%. Second, by looking at the boxplot for only one customer, one can observe that the error made varies from 4 to 22% approximately, which means that every road approximated by the distance as the crow flies can have this kind of error. The same observations can be made for the two other scenarios in Figures 3.4b and 3.4c except that now, the median of the relative error converges towards 15 and 21% respectively and the range of values is wider. Finally, it can be concluded that simply applying a correction factor to the distance as the crow flies measured is not sufficient since the error varies too much. Therefore, real time and distance by land transport are used for the rest of the report.

## 3.6 Costs of a (non-)autonomous truck

In order to make the best choice for the route followed by the truck, it is important to attribute a global cost price per delivery. These costs can be divided into three categories:

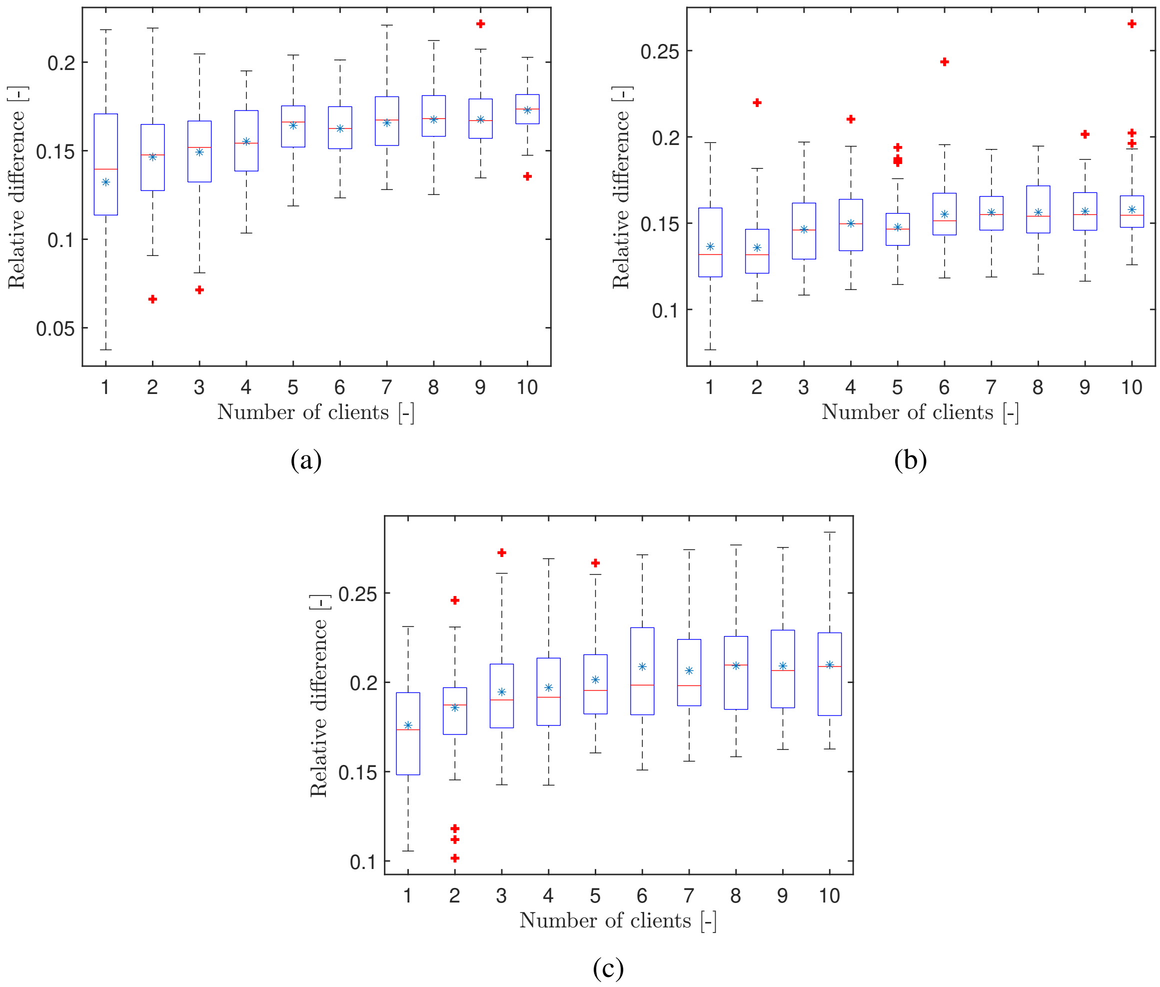



Figure 3.4 – Boxplots of the relative difference between the minimum distance by land transport and distance as the crow flies for: (a) Belgium , (b) Belgium and borders, (c) Europe

1. Variable costs, which vary with respect to the level of activity;
2. Fixed costs, which remain constant with respect to the level of activity;
3. Salary costs.

For each cost, a distinction between autonomous and non-autonomous trucks is sometimes needed as they may differ.

**Remark:** Most of these costs have already been studied in details in (Lambert, 2019) and some of them are directly used in this report.

### 3.6.1 Variable costs

The first variable cost to consider is the total fuel price which is function of the current fuel price, the consumption and the distance covered by the truck, and, therefore, the number of clients. It was already estimated in Section 3.3 that the average consumption of a heavy truck is

| [€ / km] | WALLONIA (EX VAT) | | | FLANDERS | | | BRUSSELS HIGHWAY | | |
|---|---|---|---|---|---|---|---|---|---|
| | 3.5-12 TONS | 12-32 TONS | > 32 TONS | 3.5-12 TONS | 12-32 TONS | > 32 TONS | 3.5-12 TONS | 12-32 TONS | > 32 TONS |
| Euro 0 | 0,155 | 0,208 | 0,212 | 0,122 | 0,208 | 0,234 | 0,155 | 0,208 | 0,213 |
| Euro 1 | 0,155 | 0,208 | 0,212 | 0,122 | 0,208 | 0,234 | 0,155 | 0,208 | 0,213 |
| Euro 2 | 0,155 | 0,208 | 0,212 | 0,122 | 0,208 | 0,234 | 0,155 | 0,208 | 0,213 |
| Euro 3 | 0,134 | 0,187 | 0,191 | 0,101 | 0,187 | 0,213 | 0,134 | 0,187 | 0,191 |
| Euro 4 | 0,101 | 0,154 | 0,158 | 0,068 | 0,154 | 0,180 | 0,101 | 0,154 | 0,158 |
| Euro 5 | 0,079 | 0,132 | 0,136 | 0,056 | 0,142 | 0,168 | 0,089 | 0,142 | 0,147 |
| Euro 6 | 0,079 | 0,132 | 0,136 | 0,046 | 0,132 | 0,157 | 0,079 | 0,132 | 0,136 |

Figure 3.5 – Summary of the toll due in Belgium as a function of the mass and age of the truck ©(Viappas, 2020)

0.32l/km. The fuel price considered is the one from August $12^{th}$ 2020, *i.e.* 1.38€/l, from which taxes (0.2395€/l) and excise (0.2476€/l) can be deduced, giving 0.8929€/l. Finally, the price per kilometre can be derived: **0.2857€/km**. For a better precision, a more complete model is needed and would have to take the real fuel consumption into account since in reality, a truck rolling at 70km/h or 50km/h does not have the same fuel consumption. Furthermore, the fuel price can vary from one country to another and can have a significant impact on the final cost of the total fuel price.

The second variable cost is the maintenance and repairing of the truck as well as the tires, which is directly proportional to the distance covered. The cost considered here comes from the Comité National Routier (CNR), which studies the evolution of the operational costs in the road transport industry. In normal conditions, a truck makes 113971 km/year in 228 days of use. This represents a cost of **0.026€/km** for the tires and **0.165€/km** for the maintenance and repairing (Comité National Routier, 2020a). These costs are assumed to be the same for both types of trucks even though an autonomous truck could cover a longer distance over a year.

The last variable cost is the toll, which varies across European countries. In Belgium, the toll is only due for trucks with a mass greater than 3.5 tons and depends on the year of the first release of the truck. A summary of the toll can be found in Figure 3.5 (Viappas, 2020). It is assumed that the toll is the same for both kinds of truck and therefore the difference of toll due is proportional to the price per kilometre. In our example, the price for a recent truck, Euro 6, and whose mass is between 12 and 32 tons is taken and is the same in Wallonia, Flanders and Brussels: **0.132€/km**.

Across Europe, however, it is harder to get a correct order of magnitude of the total toll since it depends on the number of kilometres done in each country. Furthermore, some countries use the Eurovignette, which is a yearly tax, including the Netherlands, Luxembourg, Denmark and Sweden and costs 1250€ for Euro 6 trucks. However, this Eurovignette tends to disappear and in the Netherlands, they are planning to abandon it and replace it by a toll by 2024. The price of the future toll is based on the current price of Belgium and Germany, around 0.15€/km (Government of the Netherlands, 2020). Finally, for the second and third case scenario, *i.e.*

Belgium and Borders and Europe, the toll used is a mean between each considered country's toll, excluding Luxembourg and taking 0.15€/km for the Netherlands for simplicity reasons. These tolls can be found in (European Commmission, 2019) for Euro 6 truck trailers. They are **0.194€/km** and **0.2855€/km** respectively.

| Type of cost | Cost |
|---|---|
| Fuel | 0.2857 €/km |
| Maintenance & repairing | 0.165 €/km |
| Tires | 0.026 €/km |
| Toll | 0.132 €/km (Belgium)<br>0.194 €/km (Belgium and borders)<br>0.2855 €/km (Europe) |
| **Total variable costs** | **0.6087 €/km (Belgium)**<br>**0.6707 €/km (Belgium and borders)**<br>**0.7622 €/km (Europe)** |

Table 3.3 – Summary of the variable costs

### 3.6.2 Fixed costs

Fixed costs are generally constant and due every year. Since it is hard to associate these costs with the distance covered or the time spent, they are expressed per day of use of the truck. The first fixed cost considered here is the depreciation of the truck which is based on its price, which comes from (Lambert, 2019). The prices suggested are 90000€ and 116100€ for the tractor units for non-autonomous and autonomous trucks respectively, for which the depreciation is made over 5 and 4 years. A quicker depreciation of the autonomous truck is assumed since it can be used more intensively due to the absence of legislation on working and driving hours. The price of the semi-trailer is 25500€ in both cases and the depreciation is done over 10 years. Assuming a linear depreciation, the yearly cost of the trucks is **20550 €/year** and **31575 €/year** respectively.

The second fixed cost is the road tax. A summary of this road tax in Wallonia can be found in (SPW Fiscalité, 2020). The value for a truck with 3 axles is used: **515 €/year** and is assumed to be the same for both trucks.

Third, the company needs an insurance for the truck as well as for the goods transported. The total price for a non-autonomous truck, **5378.79 €/year**, is based on a quote from TVM, which is an insurance company specialized in road transport and logistics. It includes several insurances:

- Third party insurance;
- Omnium insurance;

- Legal insurance;
- CMR insurance.

Again, all the details can be found in (Lambert, 2019). It is also assumed that the price for each insurance is proportional to the price of the truck and therefore, an increase of 22.6% is made for the autonomous truck, *i.e.* **6594.4 €/year**. Finally, the insurance for the goods transported, the CMR insurance, is **760.5 €/year** in both cases.

Fourth, some overhead costs must be added. They cover some costs that are not directly linked to the activity of the company but are inherent to its well being. For example: rent, furniture, electricity or heating can be considered as overhead costs. According to the Institut Transport routier et Logistique Belgique asbl, (ITLB, 2015), they account for 8 to 10% of the total yearly cost for the company. As it can be hard to estimate theses costs, it is assumed that they are equal to **15000 €/year**, which is the order of magnitude proposed by the CNR (Comité National Routier, 2020a, 2020b).

| **Type of cost** | **Cost for a non-autonomous truck** | **Cost for an autonomous truck** |
|---|---|---|
| Depreciation | 20550 €/year | 31575 €/year |
| Road tax | 515 €/year | 515 €/year |
| Truck insurances | 5378.79 €/year | 6594.4 €/year |
| CMR insurance | 760.50 €/year | 760.50 €/year |
| Overhead costs | 15000 €/year | 15000€/year |
| Days of use per year | 231 | 303 |
| **Total fixed costs per year** | 42204.29 €/year | 54444.9 €/year |
| **Total fixed costs per day** | **182.70 €/day** | **179.70 €/day** |

Table 3.4 – Summary of the fixed costs for autonomous and non-autonomous trucks

Finally, the total cost per year for each truck can be calculated. For the non-autonomous truck, it is **42204.29 €/year** and for the autonomous truck, it is **54444.9 €/year**. These prices have to be divided by the number of days the trucks are used each year in order to derive a daily price. In the case of the non-autonomous truck, it is equal to the number of days a driver works per year. If he works 5 days a week, gets 20 days off and taking the 10 legal days off in Belgium into accounts, the truck is used **231 days** per year. In the case of an autonomous truck, it can drive without ahy interruption. Assuming that the customers are generally closed on Sundays and, in any case, there is no vehicle operator to supervise it, it can drive 6 days a week. Taking into account the legal days off in Belgium, it leaves **303 days** of use per year. The daily price for using those trucks is thus **182.7 €/day** and **179.7 €/day** respectively. This is in good agreement with what is estimated by the CNR (Comité National Routier, 2020a, 2020b) for heavy trucks and long distance transport.

### 3.6.3 Salary costs

The last type of costs to be included in the model are the salary costs. Contrary to variable and fixed costs, they are expressed in Euros per hour. The salaries have already been discussed in (Lambert, 2019) which is still used as a reference here. First, for a driver, the brute salary he gets is 12.104 €/h. Employer contributions have to be added and calculated on 108% of the salary and go up to 6.777 €/hour. Finally, some extra expenses which depend on the seniority of the driver, some insurances to be taken, etc. In this case, they add up to 3.355 €/h. The total price of a driver is then: **22.24 €/h**, which is in total agreement with the estimations of the CNR (Comité National Routier, 2020a, 2020b). In addition to that, an indemnity of 38.4535€/night is given to the driver in order to cover for sleep and food expenses outside of his home, according to the Joint Committee CP 140.03 (CGSLB, 2014).

Second, the salary of a vehicle operator is estimated to be around 28 €/h, all included. However, one operator does not control only one truck but rather a fleet of trucks. There exist two ways of calculating the operator's salary affected to the truck. First, we could assume that each operator controls a fleet of $n$ vehicles and the salary affected is therefore $28/n$ €/h. The second possibility is to assume that the operator has to supervise or control the vehicle for 10 minutes when, for example, it approaches a client. The salary of the operator affected to the truck depends on the number of clients the truck manages to visit, taking the service time into account, during its delivery. Therefore, the price would be $28/n \times 10/60$ €/clients, where $n$ is the number of clients visited. The latter option is considered for the present work.

| Type of cost | Cost for a non-autonomous truck | Cost for an autonomous truck |
|---|---|---|
| Salary | 22.24 €/h | 4.67 €/clients |
| Indemnity | 38.4535 €/night | 0 €/night |

Table 3.5 – Summary of the salary costs for autonomous and non-autonomous trucks

## 3.7 Conclusion

In this section, three case scenarios were developed, from the national level, in Belgium, to the international level, in Europe. They will be compared using both the shortest and the fastest routes joining any two cities. The first option privileges a reduction of the fuel consumed as well as the greenhouse gas emissions while second option reduces salary costs. In the context of an autonomous truck, the salary costs are reduced such that choosing the fastest route may not be the optimal choice anymore. Concerning the variable costs, they should decrease when using autonomous trucks since it does not have to systematically come back to the depot as it is the case for a driver. Furthermore, the fixed costs, even though they are similar, should also decrease since the autonomous trucks is expected to complete its delivery in less days since it can drive continuously.

# 4 Optimized routes of (non-)autonomous trucks for freight transport

## 4.1 Introduction

In this section, the three case scenarios, described in Section 3.2, are studied in order to compare the optimized routes for autonomous and non-autonomous. For each case scenario, two minimizations per truck are computed:

- Non-Autonomous truck with driving Time Minimization (NATM);
- Non-Autonomous truck with driving Distance Minimization (NADM);
- Autonomous truck with driving Time Minimization (ATM);
- Autonomous truck with driving Distance Minimization (ADM);

The main difference between time and distance minimization is that the considered roads between two cities are different. In the first case, the fastest road is considered while in the second case, the shortest road is considered. These two possibilities are investigated since they would reduce the salary or the fuel expenses respectively. Each case scenario is performed 100 times for $n$ clients, and each time, the cities are selected randomly. The number of cities $n$ varies from 1 to 10. This gives a total of 4000 thousands simulations for each case scenario. Then, some data are derived, such as the cost of the delivery which depends on the number of clients, the total driving time and distance, etc. A peculiar attention is also drawn to the $\mathrm{CO_2}$ exhausted and the potential reduction obtained by using autonomous trucks. After that, a typical route taken by the trucks is analysed in details for 10 clients for each case scenario and the completion time at each client is examined in the case of driving time minimization. Finally, conclusions are drawn on whether autonomous trucks present significant advantages over non-autonomous trucks.

## 4.2 Optimized routes at the national level: Belgium

### 4.2.1 Cost analysis

The first element analysed is the price per delivery as a function of the number of clients to visit. For a better comparison, this cost is divided by the number of clients in order to know how much visiting one client costs to the company making the delivery. These costs are displayed in Figure 4.1 and shown in box plots, which offers a good summary of the distribution of the costs compared to a simple average. It can be observed that, even though the results are slightly different, the analysis is the same whether time or distance is minimized.

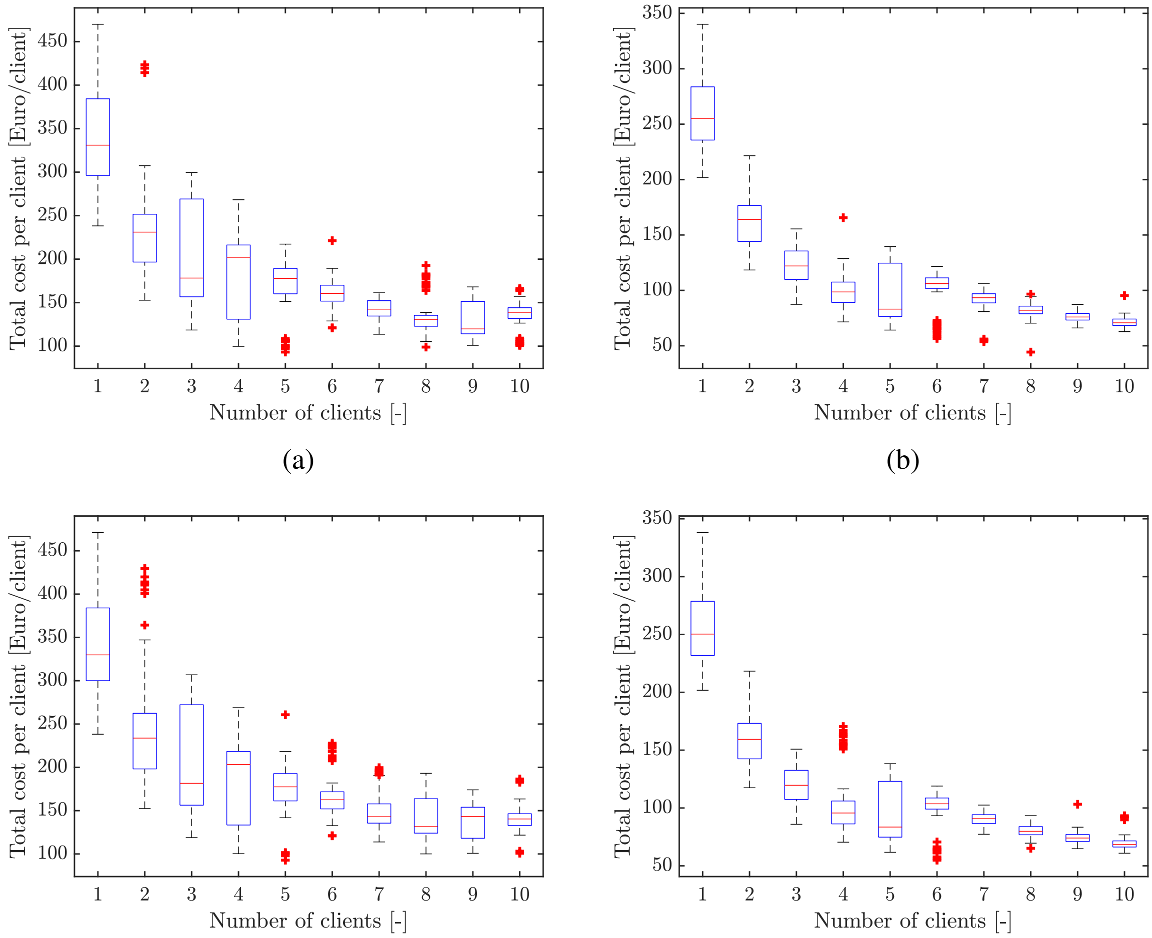
First, a global tendency, which is expected, shows that as the number of clients increases, the cost decreases. In the case of non-autonomous trucks, Figures 4.1a and 4.1c, when the number of clients is 3 or 4, a large distribution of the price is observed. This is explained by the fact that this is the moment when a second truck is needed and therefore drastically increases the part of the fixed costs on the cost per client if two trucks are used. The same phenomena appears for 8 and 9 clients when a third truck might be needed. In the case of autonomous trucks, Figures 4.1b and 4.1d, the key number of clients is 5, which corresponds to the moment the truck has to be used for a second consecutive day to complete the delivery, involving also a sudden increase of the fixed costs. This tendency can already be observed for 4 clients since there are quite a few outliers with a high cost.

Second, concerning the values of these costs, the autonomous truck presents a huge advantage over non-autonomous trucks. In fact, if only one client has to be delivered, the median price for non-autonomous trucks is around 330€ while it is around 250€ for autonomous trucks. The price decreases by 20 to 26% if the $75^{th}$ and $25^{th}$ percentiles are considered. In the case of 10 clients, the tendency is enhanced since the price decreases from around 140€ to 70€. There is a 50% decrease. The same observations can be made if distance is minimized. However, it is important to keep in mind all the hypotheses that have been made in section 3.6 in the case of an autonomous truck, whether it be for the price of the tractor or the salary of the vehicle operator. They all have an influence on the final results which may differ from reality.

The second element analysed is the distance covered and the time spent on the road, in Figures 4.2a and 4.2b respectively. As assumed in Section 3.3, the distance covered is directly proportional to the $CO_2$ exhausted by the truck. In fact, for each kilometre, 843g of $CO_2$ are produced. In the case of an autonomous truck, if its capacity allows it to visit all the clients, the total distance is less than for non-autonomous trucks since several trucks are needed and all of them have to start from and go back to the depot in Brussels. There could be a reduction of 19% of the $CO_2$ gases if time is minimized, or even 25% if distance is minimized. It can also be observed that in certain cases, minimizing the distance over speed for non-autonomous trucks leads to a higher total distance, which is counter intuitive. This can be explained by the fact that since the drivers can only work 9 hours, an additional truck can be required when minimizing distance while it might be not necessary if time is minimized, and therefore, a new depot-customer route and a new customer-depot route are added.

Concerning the driving time, using autonomous trucks could reduce it by 21% which could in turn decrease traffic jam if the same amount of deliveries are made. In reality, though, there would be more trucks available, since it takes less time per delivery, and they would be cheaper, which could induce an increase in the demand and, maybe, more trucks would be on the roads. This is illustrated by a simple supply and demand model in the case of an increase of the supply in Figure 4.3.

The relative cost difference for autonomous trucks when minimizing distance or time is very little and is discussed any further. However, this relative cost difference is shown in Figure B.2 for the interested reader.




(a) (b) (c) (d)

Figure 4.1 – Total cost per client of the trucks used in Belgium: (a) NATM, (b) ATM, (c) NADM and (d) ADM.

### 4.2.2 Route analysis

Comparing the routes of non-autonomous and autonomous trucks is also very interesting. The case considered here is time minimization with 10 clients: 4 are in the South of Brussels, 5 in the North and 1 on the West. Three non-autonomous trucks are needed to complete the delivery and their journey is detailed in Figure 4.4a. While the route of the second truck, in orange, is rather classical, the one of the first truck, in blue, is intriguing. The roads linking the different clients cross each other while in general, this is not the case. Therefore, one could expect the following sequence: $1 \to 10 \to 38 \to 14 \to 5 \to 1$. However, the following sequence is observed: $1 \to 10 \to 14 \to 38 \to 5 \to 1$. This can be explained by the fact that at the intersection of the roads $10 \to 14$ and $38 \to 5$, there is the Daussoulx highway interchange and many roads can go through it, which is also the case for the road $10 \to 38$. In the present case, it

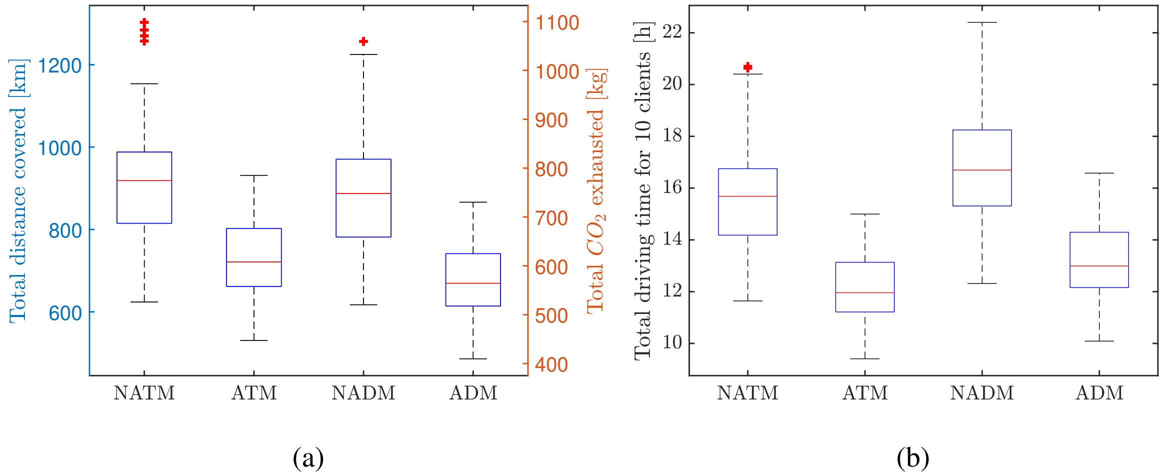



Figure 4.2 – Box plots of the distribution of (a) the total distance covered and (b) the total driving time in Belgium for 10 clients

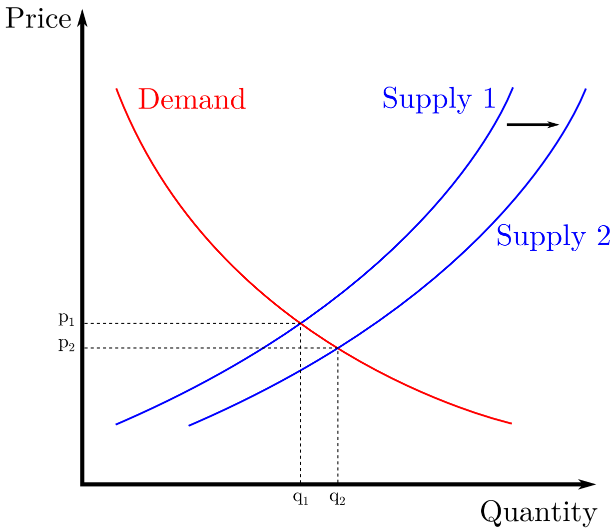



Figure 4.3 – Supply and demand model in the case of an increase of the supply

is quicker by doing as shown in Figure 4.4a since these are straight roads and no time is lost in the highway interchange. Finally, the last truck is only used to visit one client, which may not be a good option for the delivery company. Concerning the autonomous case, in Figure 4.4b, only one truck is needed and the same remark concerning the Daussoulx highway interchange can be made. Here, however, the final completion time, *i.e.* when the last client is visited, is on the second day, while everything could have been done in one day in the non-autonomous case, providing that three vehicles are indeed available.

## 4.3 Optimized routes at the international level: Belgium and borders

### 4.3.1 Cost analysis

The evolution of the distribution of the costs with respect to the number of clients to visit and for all types of minimization is shown in Figure 4.5. First, it can be observed that in all cases, the evolution of the cost per client has the same behaviour. This can be attributed to the fact that since the driver can sleep outside and drive for 56 hours per week then, most of the time, only one truck is needed for the whole delivery, even for 10 clients. Therefore, the truck visits all the clients without going back to the depot several times, which reduce the fuel consumption and its price.

The main difference in price between autonomous and non-autonomous trucks is the number of times the fixed costs have to be taken into account. In fact, since the clients are open for 16 hours, the autonomous truck can visit much more clients in one day compared to a driver who can only drive for 9 hours per day. Therefore, the final completion time for the last client can be much sooner for autonomous trucks, for example one or two days earlier. Another difference, but to a lesser extent, is the indemnities given to the driver to cover for his food and sleep expenses.

Concerning the gain obtained per client by using autonomous trucks, it is quite similar to the first case scenario. For one client, the costs are reduced by 22 to 27% while it is about 42% for 10 clients. It can even go up to 47% for 6 clients. These differences mainly come from the fixed costs of the trucks but also on how the driver's of the vehicle operator's salary is affected to the truck.

Finally, it can be observed in Figure 4.6a that if the driver is allowed to sleep during his journey, then the routes taken by autonomous and non-autonomous trucks are the same and therefore, the total distance covered and the $CO_2$ exhausted are also the same. Obviously, this would not be the case if drivers had to come back to the depot every day. In fact, some customers are approximately 4.5 hours away from the depot and the driver could only visit one of them per day, which would result in an increase of vehicles used. Such an analysis has not been done since it requires too much computational time. Concerning the driving time, in Figure 4.6b, it is also the same in both situations, but the repartition of the driving time over the day for an autonomous truck would be more homogeneous and there would thus be less trucks during peak time. Furthermore, as already explained, even though the driving time is the same, the completion time is earlier for autonomous trucks. This is emphasized in the next section where a typical delivery for ten clients is analysed.

### 4.3.2 Route analysis

The delivery analysed here is for 10 clients mainly located in France, Germany and the Netherlands and driving time is minimized. First, as it can be seen in Figures 4.7a and 4.7b, the routes taken by both trucks are exactly the same, confirming the observation made from Figure 4.6. Second, in the case of the non-autonomous truck, 3 days are required for the delivery and therefore the fixed costs are higher than for an autonomous truck since it requires only 2 days. Finally, depending on when the driver starts his journey (here, he starts such that he arrives at the opening hour of the first client), the driver could spend most of his driving time during the day, from 7am to 4pm for example, while the autonomous truck can drive any time, freeing the roads at peak time.

## 4.4 Optimized routes at the international level: continental Europe

This scenario is the one that reveals all the potential of the autonomous trucks. Again, a cost analysis is made before a typical route analysis.

**Remark:** for this scenario, the routing API for the shortest road did not work very well and often, the matrix API was used instead without information on how the *best* road is computed. However, most of the time, the road corresponds to the fastest road. Therefore, the results concerning the distance minimization may not be exploitable but are still displayed.

### 4.4.1 Cost analysis

The costs per client are displayed in Figures 4.8a and 4.8b for non-autonomous and autonomous trucks respectively in the case of time minimization. For only one client to deliver, which is a scenario that can happen in the case of a full truck load, the expenses decrease by 40% when using autonomous trucks for the whole cost distribution. For example, the median cost is 3118€ for a non-autonomous truck while it is only 1860€ for an autonomous truck. As the number of clients increases, the cost per client decreases but the distribution is still wide for non-autonomous trucks. This is due to the fact that depending on the clients to visit, the journey can take 2 or 6 days, with all the costs that come along. For 10 clients, the decrease is around 55% and can even go up to 60%. As expected, this scenario reveals all the potential of the autonomous trucks.

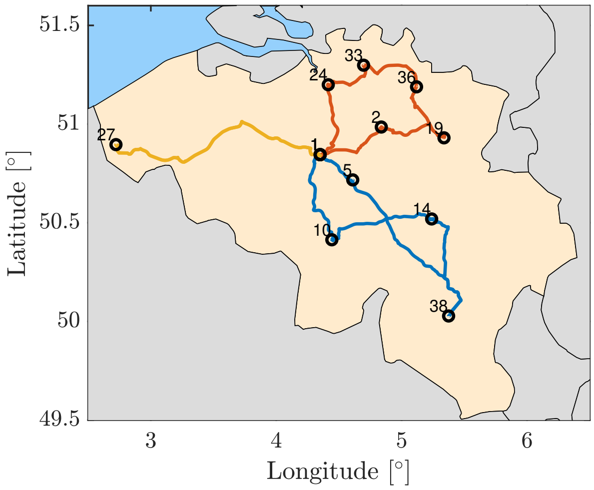
Concerning the total distance and thus the $CO_2$, it could decrease by 14 to 28% for 10 clients, as it can be seen in Figure 4.9a. Concerning the total driving time, it would also decrease by 14 to 28%, but again, it is also important to look at the final completion time of a typical route to better understand the benefits of the autonomous trucks. For example, it could take 5 days for the autonomous truck to complete the delivery, and the same time for non-autonomous trucks, but in the latter case, 3 trucks would be needed. And again, autonomous trucks could even stop during peak time.

### 4.4.2 Route analysis

Again, the route optimization with 10 clients to visit is done in the case of autonomous and non-autonomous trucks with time minimization. Both routes are displayed in Figure 4.8a and 4.4b respectively. The first thing to observe in the cities to be visited is that one of them, Lisbon (19), is isolated from the others and far away from the depot as it takes approximately 26 hours by truck to go there. Therefore, when using non-autonomous trucks, we need one vehicle for a full week to do the round trip. By chance, client 9 is on the way and can be visited during the journey, but no one else could be added as it requires 52 driving hours for the journey. Second, 3 trucks are needed to complete the full delivery and each one of them takes 5 or 6 days to visit a few clients. On the contrary, when using an autonomous truck, the client in Lisbon is integrated in a single journey which visits all the clients in one trip. Therefore, the autonomous truck manages to go back to the depot by Thursday 7pm.

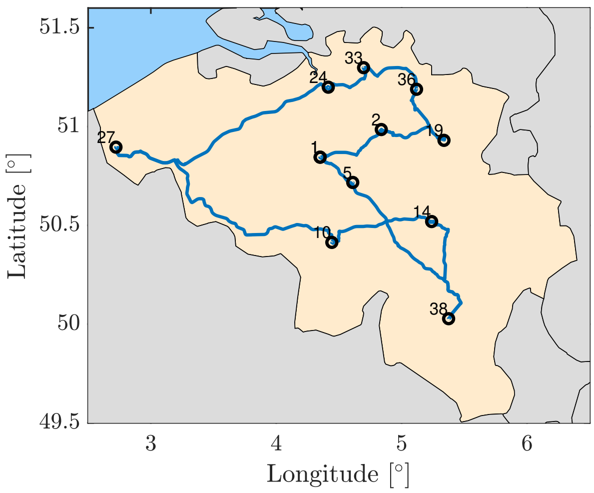
## 4.5 Conclusion

It was shown in this section that using autonomous trucks drastically reduces the cost per client for each delivery by at least 20% when only one client is considered and up to 50 or 60% when several clients have to be delivered and for each case scenario. In addition to these cost reductions, a huge reduction of the number of vehicle used was also highlighted whether it be for national transport or very long distance transport. Less trucks used would mean a less dense traffic especially at peak time. Furthermore, since the autonomous trucks do not have to systematically come back to the depot after a certain number of hours, the total distance covered as well as the greenhouse gas emissions would reduce by approximately 14 to 28% in the case of the European and Belgian case scenarios. More details about the relative cost difference between autonomous and non-autonomous trucks for each case scenario when time is minimized can be found in Figure B.1.




| Vehicle | Stop | Arrival | Departure |
|---|---|---|---|
| 1 | 1 | - | 06:56 |
| | 10 | 8:00 | 8:30 |
| | 14 | 09:29 | 09:59 |
| | 38 | 11:06 | 11:36 |
| | 5 | 13:52 | 14:22 |
| | 1 | 15:05 | - |
| 2 | 1 | - | 07:00 |
| | 24 | 8:00 | 8:30 |
| | 33 | 09:02 | 09:32 |
| | 36 | 10:24 | 10:54 |
| | 19 | 11:43 | 12:13 |
| | 2 | 13:01 | 13:31 |
| | 1 | 15:15 | - |
| 3 | 1 | - | 05:47 |
| | 27 | 8:00 | 8:30 |
| | 1 | 10:43 | - |

(a)




| Vehicle | Stop | Arrival | Departure |
|---|---|---|---|
| 1 | 1 | - | Mon 07:17 |
| | 5 | Mon 8:00 | Mon 8:30 |
| | 38 | Mon 10:01 | Mon 10:31 |
| | 14 | Mon 11:38 | Mon 12:08 |
| | 10 | Mon 13:07 | Mon 13:37 |
| | 27 | Mon 15:57 | Mon 16:27 |
| | 24 | Tue 08:00 | Tue 08:30 |
| | 33 | Tue 09:02 | Tue 09:32 |
| | 36 | Tue 10:24 | Tue 10:54 |
| | 19 | Tue 11:43 | Tue 12:13 |
| | 2 | Tue 13:01 | Tue 13:31 |
| | 1 | Tue 14:30 | - |

(b)

Figure 4.4 – Comparison of the routes taken by the trucks in Belgium: (a) NATM, (b) ATM

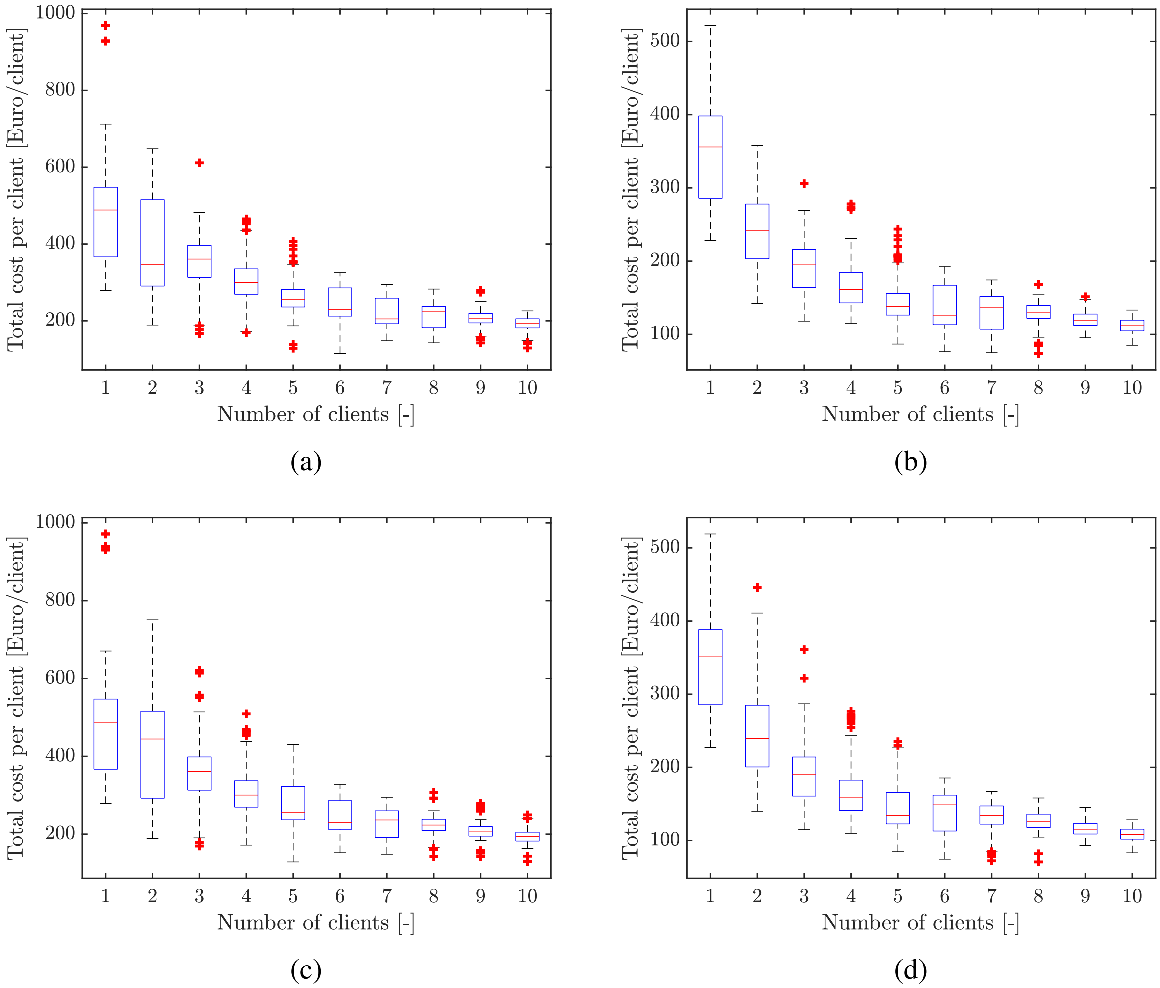



Figure 4.5 – Total cost per client of the trucks used in Belgium and borders: (a) NATM, (b) ATM, (c) NADM and (d) ADM.

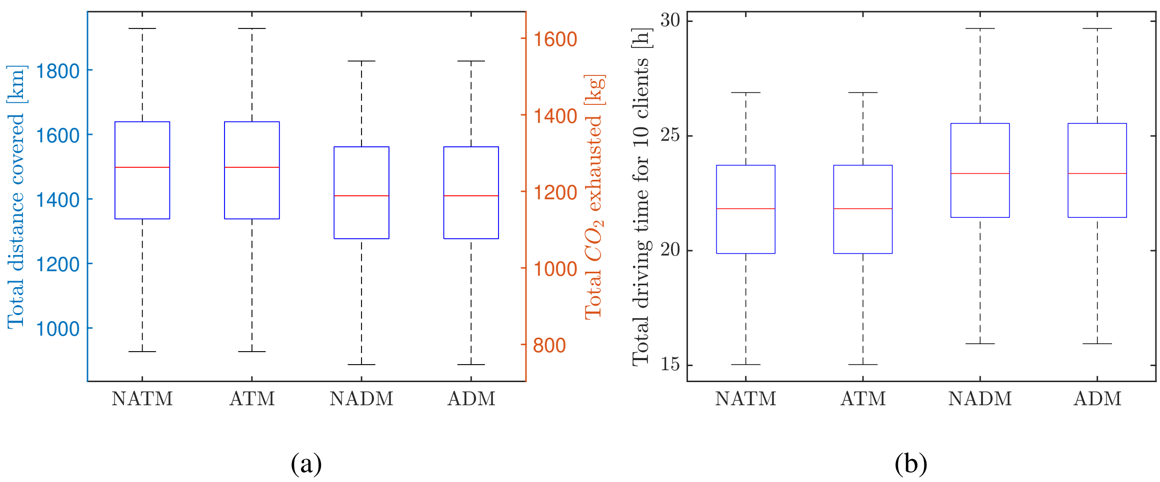
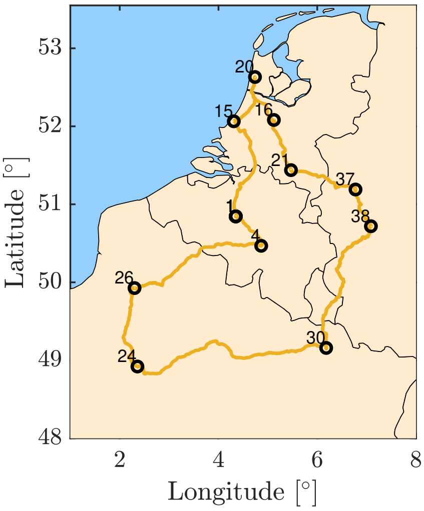
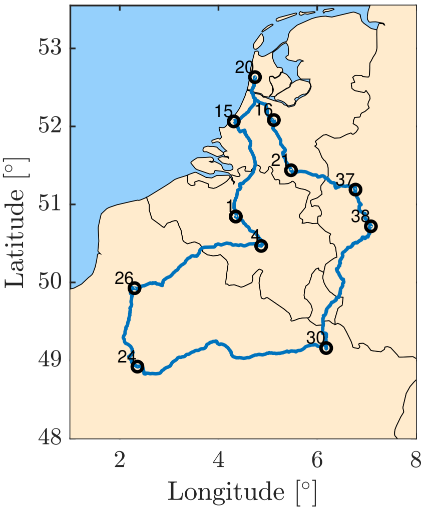



Figure 4.6 – NFRC of the fundamental resonance of the Duffing oscillator for varying forcing amplitudes: (a) Amplitude/Frequency and (b) Phase lag/Frequency




| Vehicle | Stop | Arrival | Departure |
|---|---|---|---|
| 1 | 1 | - | Mon 04:48 |
| | 4 | Mon 06:00 | Mon 06:30 |
| | 26 | Mon 09:33 | Mon 10:03 |
| | 24 | Mon 12:41 | Mon 13:11 |
| | 30 | Tue 07:31 | Tue 08:01 |
| | 38 | Tue 12:16 | Tue 12:46 |
| | 37 | Tue 13:55 | Tue 14:25 |
| | 21 | Tue 16:11 | Tue 16:41 |
| | 16 | Wed 07:58 | Wed 08:28 |
| | 20 | Wed 09:40 | Wed 10:10 |
| | 15 | Wed 11:22 | Wed 11:52 |
| | 1 | Wed 15:14 | - |

(a)




| Vehicle | Stop | Arrival | Departure |
|---|---|---|---|
| 1 | 1 | - | Mon 04:48 |
| | 4 | Mon 6:00 | Mon 6:30 |
| | 26 | Mon 09:33 | Mon 10:03 |
| | 24 | Mon 11:56 | Mon 12:56 |
| | 30 | Mon 16:46 | Mon 17:16 |
| | 38 | Mon 20:46 | Mon 21:16 |
| | 37 | Tue 06:00 | Tue 06:30 |
| | 21 | Tue 08:16 | Tue 08:46 |
| | 16 | Tue 10:03 | Tue 10:33 |
| | 20 | Tue 11:45 | Tue 12:15 |
| | 15 | Tue 13:27 | Tue 13:57 |
| | 1 | Tue 16:34 | - |

(b)

Figure 4.7 – Comparison of the routes taken by the trucks in Belgium and borders: (a) NATM, (b) ATM

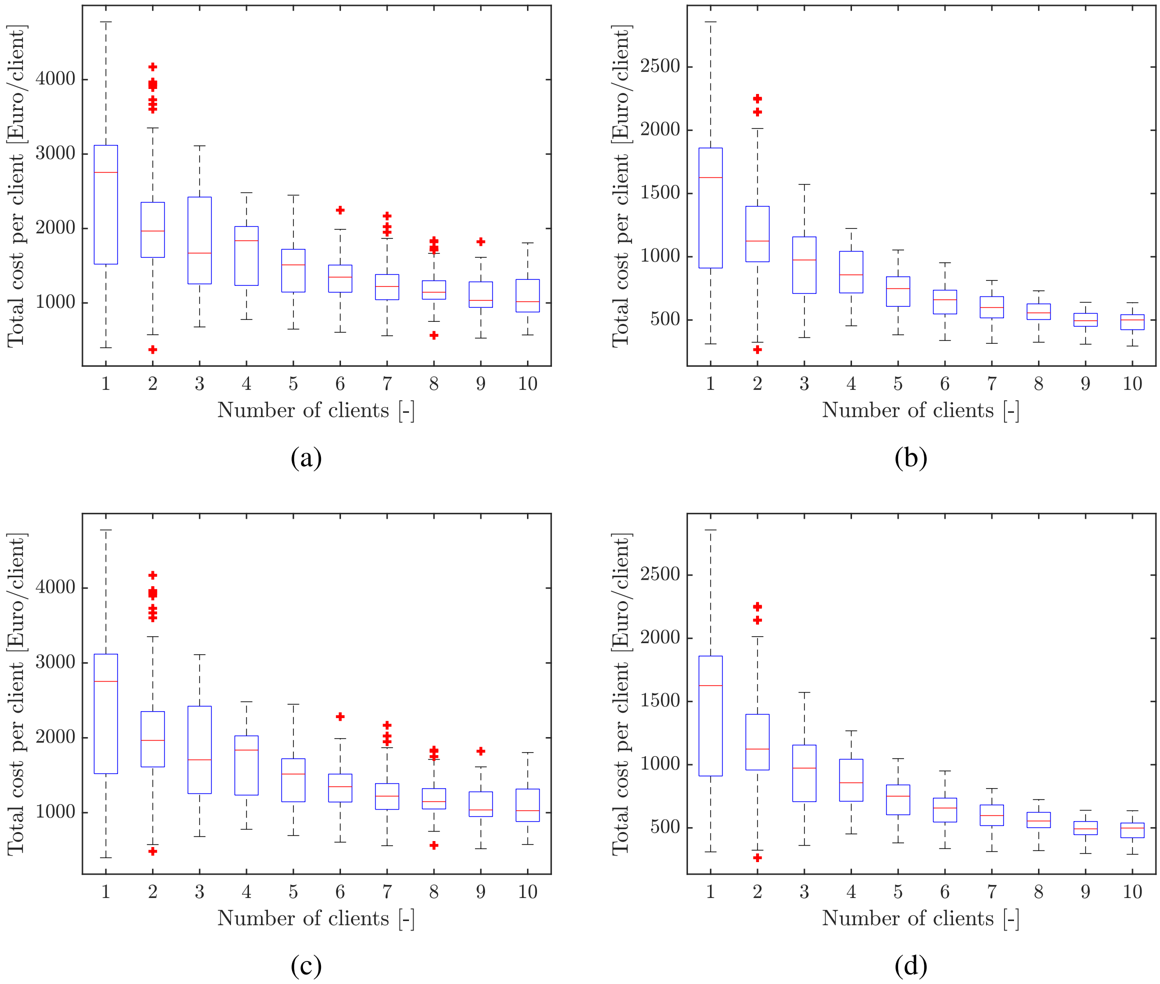



Figure 4.8 – Total cost per client of the trucks used in Europe: (a) NATM, (b) ATM, (c) NADM and (d) ADM.

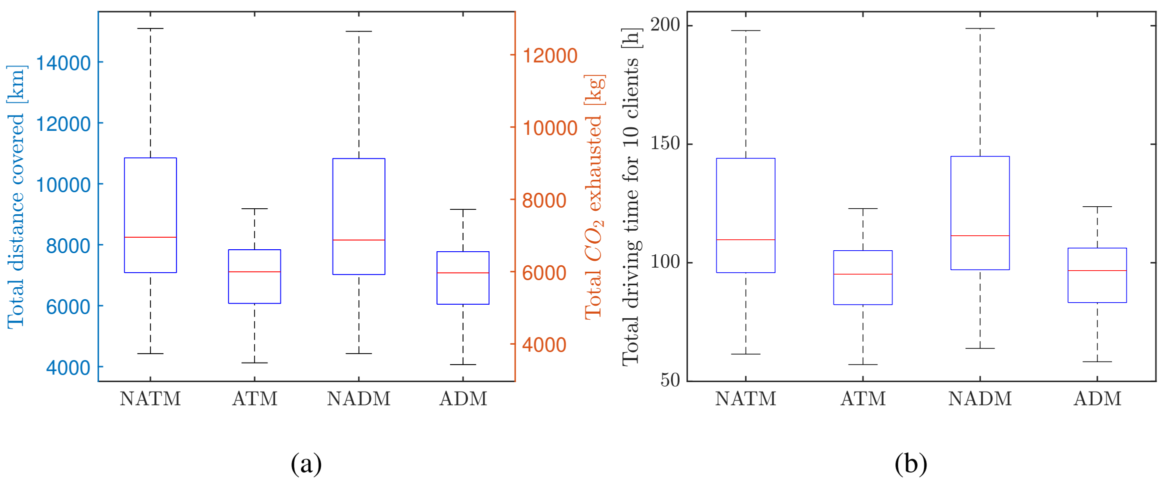
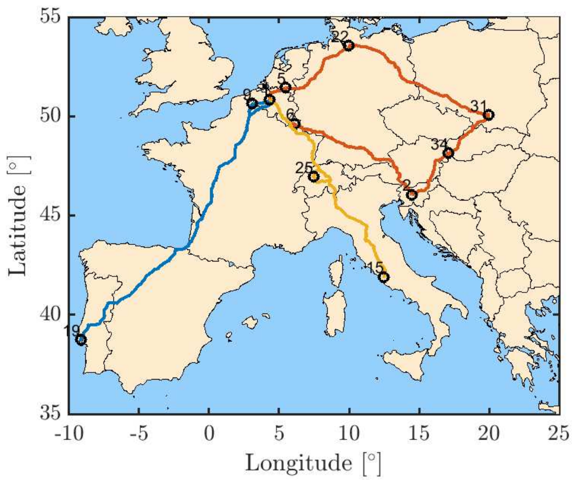
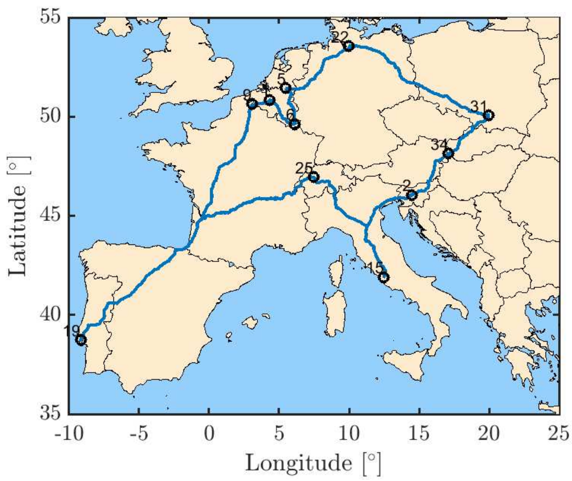



Figure 4.9 – NFRC of the fundamental resonance of the Duffing oscillator for varying forcing amplitudes: (a) Amplitude/Frequency and (b) Phase lag/Frequency




| Vehicle | Stop | Arrival | Departure |
|---|---|---|---|
| 1 | 1 | - | Mon 00:00 |
| | 9 | Mon 01:45 | Mon 02:15 |
| | 19 | Wed 09:26 | Wed 09:56 |
| | 1 | Sat 08:20 | - |
| 2 | 1 | - | Mon 00:00 |
| | 6 | Mon 03:05 | Mon 03:35 |
| | 2 | Tue 07:09 | Tue 07:39 |
| | 34 | Wed 03:18 | Wed 03:48 |
| | 31 | Wed 10:21 | Wed 10:51 |
| | 22 | Fri 03:14 | Fri 03:44 |
| | 5 | Fri 10:41 | Fri 11:11 |
| | 1 | Sat 03:19 | - |
| 3 | 1 | - | Mon 00:00 |
| | 15 | Wed 00:37 | Wed 01:07 |
| | 25 | Thu 03:42 | Thu 04:12 |
| | 1 | Fri 03:30 | - |

(a)




| Vehicle | Stop | Arrival | Departure |
|---|---|---|---|
| 1 | 1 | - | Mon 00:00 |
| | 9 | Mon 01:45 | Mon 02:15 |
| | 19 | Tue 03:11 | Tue 03:41 |
| | 25 | Tue 05:20 | Tue 05:50 |
| | 15 | Tue 17:41 | Tue 18:11 |
| | 2 | Wed 03:54 | Wed 04:24 |
| | 34 | Wed 10:03 | Wed 10:33 |
| | 31 | Wed 16:21 | Wed 16:51 |
| | 22 | Thu 04:30 | Thu 05:00 |
| | 5 | Thu 11:11 | Thu 11:41 |
| | 6 | Thu 15:25 | Thu 15:55 |
| | 1 | Thu 19:00 | - |

(b)

Figure 4.10 – Comparison of the routes taken by the trucks in Europe: (a) NATM, (b) ATM

# 5 Conclusion and future work

## 5.1 Conclusion

Many problems concerning the comparison between autonomous and non-autonomous trucks were tackled in this work.

First, an efficient implementation of the VRP which gives an optimal solution was done. This problem had to include the European legislation on driving and working hours, which made the problem very deterministic. The choice of dynamic programming was made since it is flexible and gives the best computational time for obtaining an optimal solution. Then, in order to make the VRP realistic, real routes using road transport, using an API, were computed and compared with straight line roads. Using the distance as the crow flies underestimates the real distance by around 5 to 20% which can not be neglected. Finally, for each route a cost had to be associated. These costs were divided into three categories: variable costs, fixed costs and salary costs. A summary of theses costs can be found in table 5.1. Even though variable costs and fixed costs are similar for both trucks, they strongly depend on the final completion time and the total distance covered. In fact, a driver might have to go back to the depot before completing the delivery, increasing the costs, while the autonomous truck can wait anywhere before the next client's opening.

| Type of cost | Cost for a non-autonomous truck | Cost for an autonomous truck |
|---|---|---|
| Variable costs | 0.6087 €/km (Belgium)<br>0.6707 €/km (Belgium and borders)<br>0.7622 €/km (Europe) | 0.6087 €/km (Belgium)<br>0.6707 €/km (Belgium and borders)<br>0.7622 €/km (Europe) |
| Fixed costs | 182.70 €/day | 179.70 €/day |
| Salary costs | 22.24 €/h<br>38.4535 €/night (indemnity) | 4.67 €/clients<br>0 €/night (indemnity) |

Table 5.1 – Summary of the costs for both autonomous and non autonomous trucks

Three case scenarios were considered. For the first one, in Belgium, the driver could only drive or work 9 hours per day in order to sleep at his place and the clients were only open from 8am to 5pm, leaving a short time window to be delivered. The second scenario extends the first scenario to the borders of Belgium by adding the Netherlands, Luxembourg, the North of France and the West of Germany. In this scenario, the driver is allowed to sleep on the roads in exchange for indemnities that covers for his sleep and food expenses. In this case, the clients are open from 6am to 10pm even though the drivers can only drive for maximum 9 hours per day due to the European legislations. The last scenario is for long distance transport across continental Europe and the client plays the role of transport hubs which are open 24 hours a day. Each case scenario was computed 100 times for $n$ cities chosen randomly and $n$ going from 1 to 10.

The results speak for themselves. For the first case scenario, the cost per delivery decreases by 20 to 50% for 1 and 10 clients respectively. Concerning the final completion, from 5 to 10 clients, the autonomous trucks need 2 days to complete the delivery while everything could be done in one day for regular trucks providing 2 or 3 trucks and drivers are available. Furthermore, greenhouse gas emissions are reduced by around 20% which is a first step towards the Green Deal goal which is to make Europe neutral in greenhouse gas emissions by 2050. For the second case scenario, the same cost reductions and final completion time as the first scenario are found . However, the $CO_2$ do not reduce since the driver is able to sleep outside and therefore generally takes the same route as the autonomous truck. Finally, for the third case scenario, the reduction costs are enhanced and vary between 40% for 1 client 55% for 10 clients and even up to 60% for 6 clients. A typical route for 10 clients would need three drivers from Monday to Friday or Saturday while with only one autonomous truck, the last client would be delivered on Thursday. Furthermore, there could be a reduction of up to 28% of the greenhouse gas emissions.

## 5.2 Future work

In this work, the whole comparison between autonomous and non autonomous trucks was based on three case scenarios for the VRP with Time Windows and assumed similar trucks that are fuel-powered. Furthermore, strong assumptions were made, especially for the costs. For example, the price of the autonomous truck could vary from one application to another, as well as the price for the insurance. In addition to that, the salary of the vehicle operator affected to the truck was proportional to the number of clients visited. Other choices could have been made and thus other case scenarios could have been explored.

For example, the present work compared two fuel-powered trucks. in reality though, electric trucks are introduced into the market. The price of the electricity is not the same as the fuel and the charging time should also be taken into account since it can take more than 30 minutes. The range per charge is also smaller than a filled in fuel-powered truck. This work does not take this aspect into account and it could be interesting to do so.

Furthermore, only the Time Windows variant of the VRP was considered here. Adding capacity constraints could also be very interesting, although it might increase the time complexity of the problem, which was already a small issue here.

Finally, it could also be interesting to compare different types of autonomous trucks for other specific applications, such as the transport of people, home deliveries of food in a city. Many applications exist and as many analyses can be done.

# A Longitude and latitude of visited cities

| N° | City | Latitude | Longitude |
|---|---|---|---|
| 1 | Brussels | 50.84503 | 4.34993 |
| 2 | Ljubljana | 46.04552 | 14.46168 |
| 3 | Zagreb | 45.79486 | 15.97907 |
| 4 | Amsterdam | 52.36915 | 4.89238 |
| 5 | Eindhoven | 51.43832 | 5.46851 |
| 6 | Luxembourg | 49.61058 | 6.12896 |
| 7 | Paris | 48.85653 | 2.33576 |
| 8 | Montpellier | 43.62430 | 3.86181 |
| 9 | Lille | 50.62792 | 3.06144 |
| 10 | Brest | 48.39015 | -4.48613 |
| 11 | Madrid | 40.42064 | -3.70707 |
| 12 | Barcelone | 41.39442 | 2.17588 |
| 13 | Seville | 37.39078 | -5.98448 |
| 14 | Valence | 39.47251 | -0.37584 |
| 15 | Rome | 41.89812 | 12.48738 |
| 16 | Milan | 45.44074 | 9.17080 |
| 17 | Naples | 40.85272 | 14.26771 |
| 18 | Porto | 41.15593 | -8.62959 |
| 19 | Lisbonne | 38.72555 | -9.15039 |
| 20 | Berlin | 52.51832 | 13.40696 |
| 21 | Munich | 48.13901 | 11.57305 |
| 22 | Hambourg | 53.55196 | 9.99454 |
| 23 | Cologne | 50.93752 | 6.95403 |
| 24 | Zurich | 47.37567 | 8.53918 |
| 25 | Berne | 46.94875 | 7.44767 |
| 26 | Vienne | 48.20982 | 16.37271 |
| 27 | Graz | 47.07140 | 15.43941 |
| 28 | Prague | 50.08951 | 14.44297 |
| 29 | Brno | 49.19529 | 16.60725 |
| 30 | Varsovie | 52.22821 | 20.95719 |
| 31 | Cracovie | 50.06403 | 19.94485 |
| 32 | Gdansk | 54.25259 | 18.64730 |
| 33 | Budapest | 47.48504 | 19.02463 |
| 34 | Bratislava | 48.13617 | 17.10392 |
| 35 | Kosice | 48.72079 | 21.27364 |

Table A.1 – Cities visited in Europe

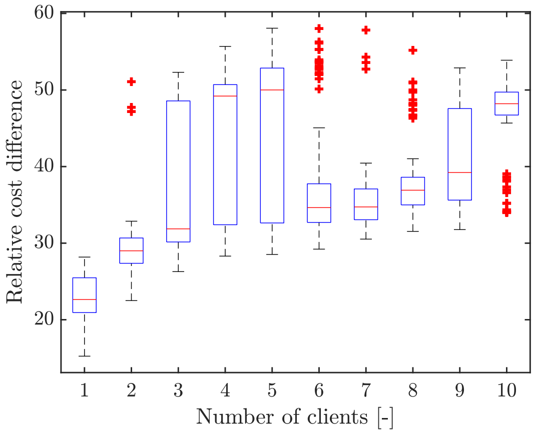
| N° | City | Latitude | Longitude |
|---|---|---|---|
| 1 | Brussels | 50.8450 | 4.3499 |
| 2 | Aarschot | 50.9850 | 4.8376 |
| 3 | Ostende | 51.2176 | 2.9130 |
| 4 | Waremme | 50.6976 | 5.2559 |
| 5 | Wavre | 50.7171 | 4.6100 |
| 6 | Louvain-La-Neuve | 50.6688 | 4.6099 |
| 7 | Arlon | 49.6815 | 5.8081 |
| 8 | Nivelles | 50.5951 | 4.3249 |
| 9 | Mons | 50.4487 | 3.9481 |
| 10 | Charleroi | 50.4137 | 4.4433 |
| 11 | Namur | 50.4651 | 4.8647 |
| 12 | Yvoir | 50.3275 | 4.8784 |
| 13 | Ciney | 50.2975 | 5.1008 |
| 14 | Huy | 50.5192 | 5.2392 |
| 15 | Rochefort | 50.1788 | 5.2204 |
| 16 | Val-Dieu | 50.7014 | 5.8144 |
| 17 | Achouffe | 50.1508 | 5.7460 |
| 18 | Orval | 49.6383 | 5.3498 |
| 19 | Hasselt | 50.9303 | 5.3375 |
| 20 | Chimay | 49.9812 | 4.3368 |
| 21 | Hal | 50.7118 | 4.2373 |
| 22 | Buggenhout | 51.0135 | 4.2015 |
| 23 | Bruges | 51.2069 | 3.2259 |
| 24 | Anvers | 51.1993 | 4.4149 |
| 25 | Sourbrodt | 50.4863 | 6.1098 |
| 26 | Gouvy | 50.2149 | 5.9130 |
| 27 | Westvleteren | 50.8959 | 2.7218 |
| 28 | Achel | 51.2982 | 5.4889 |
| 29 | Hoegaarden | 50.7730 | 4.9028 |
| 30 | Melle | 51.0004 | 3.8053 |
| 31 | Audenarde | 50.8455 | 3.6181 |
| 32 | Vichte | 50.8381 | 3.4042 |
| 33 | Malle | 51.2982 | 4.6964 |
| 34 | Alken | 50.8776 | 5.3061 |
| 35 | Turnhout | 51.3224 | 4.9430 |
| 36 | Mol | 51.1887 | 5.1188 |
| 37 | Bouillon | 49.7962 | 5.0688 |
| 38 | Saint-Hubert | 50.0275 | 5.3745 |
| 39 | Spa | 50.4913 | 5.8639 |
| 40 | Genk | 50.9656 | 5.5030 |

Table A.2 – Cities visited in Belgium

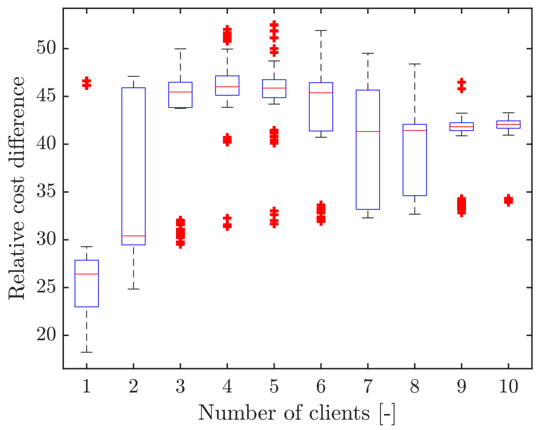
| N° | City | Latitude | Longitude |
|---|---|---|---|
| 1 | Brussels | 50.8450 | 4.3499 |
| 2 | Liege | 50.6349 | 5.5689 |
| 3 | Mons | 50.4543 | 3.9523 |
| 4 | Namur | 50.4678 | 4.8650 |
| 5 | Charleroi | 50.4117 | 4.4446 |
| 6 | Arlon | 49.6842 | 5.8146 |
| 7 | Anvers | 51.2184 | 4.4035 |
| 8 | Bruges | 51.2087 | 3.2248 |
| 9 | Hasselt | 50.9302 | 5.3374 |
| 10 | Aarschot | 50.9849 | 4.8361 |
| 11 | Genk | 50.9656 | 5.5030 |
| 12 | Maastricht | 50.8480 | 5.6893 |
| 13 | Groningue | 53.2172 | 6.5520 |
| 14 | Amsterdam | 52.3691 | 4.8924 |
| 15 | La Haye | 52.0643 | 4.3102 |
| 16 | Utrecht | 52.0814 | 5.1220 |
| 17 | Arnhem | 51.9860 | 5.9052 |
| 18 | Rotterdam | 51.9231 | 4.4795 |
| 19 | Leeuwarden | 53.1974 | 5.8013 |
| 20 | Alkmaar | 52.6321 | 4.7396 |
| 21 | Eindhoven | 51.4383 | 5.4685 |
| 22 | Luxembourg | 49.6106 | 6.1290 |
| 23 | Clervaux | 50.0539 | 6.0300 |
| 24 | Saint-Denis | 48.9205 | 2.3584 |
| 25 | Rouen | 49.4889 | 1.1415 |
| 26 | Amiens | 49.9264 | 2.2999 |
| 27 | Reims | 49.2845 | 3.9784 |
| 28 | Lille | 50.6051 | 3.1011 |
| 29 | Calais | 50.9376 | 1.9041 |
| 30 | Metz | 49.1587 | 6.1811 |
| 31 | Saint-Quentin | 49.8213 | 3.3025 |
| 32 | Sedan | 49.7039 | 4.9202 |
| 33 | Dunkerke | 51.0229 | 2.3998 |
| 34 | Dortmund | 51.4973 | 7.4193 |
| 35 | Aachen | 50.7504 | 6.0767 |
| 36 | Cologne | 50.9333 | 7.0002 |
| 37 | Dusseldorf | 51.1897 | 6.7755 |
| 38 | Bonn | 50.7191 | 7.0872 |
| 39 | Monchengladbach | 51.1810 | 6.4027 |
| 40 | Duren | 50.7955 | 6.4617 |
| 41 | Duisbourg | 51.4452 | 6.8023 |
| 42 | Wuppertal | 51.2466 | 7.0955 |
| 43 | Heinsberg | 51.0539 | 6.1057 |

Table A.3 – Cities visited in Belgium and borders

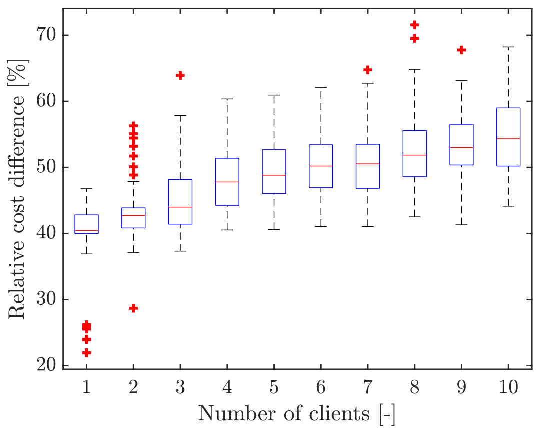
# B Relative cost differences




(a)




(b)




(c)

Figure B.1 – Boxplots of the relative cost difference between NATM and ATM for: (a) Belgium , (b) Belgium and borders, (c) Europe

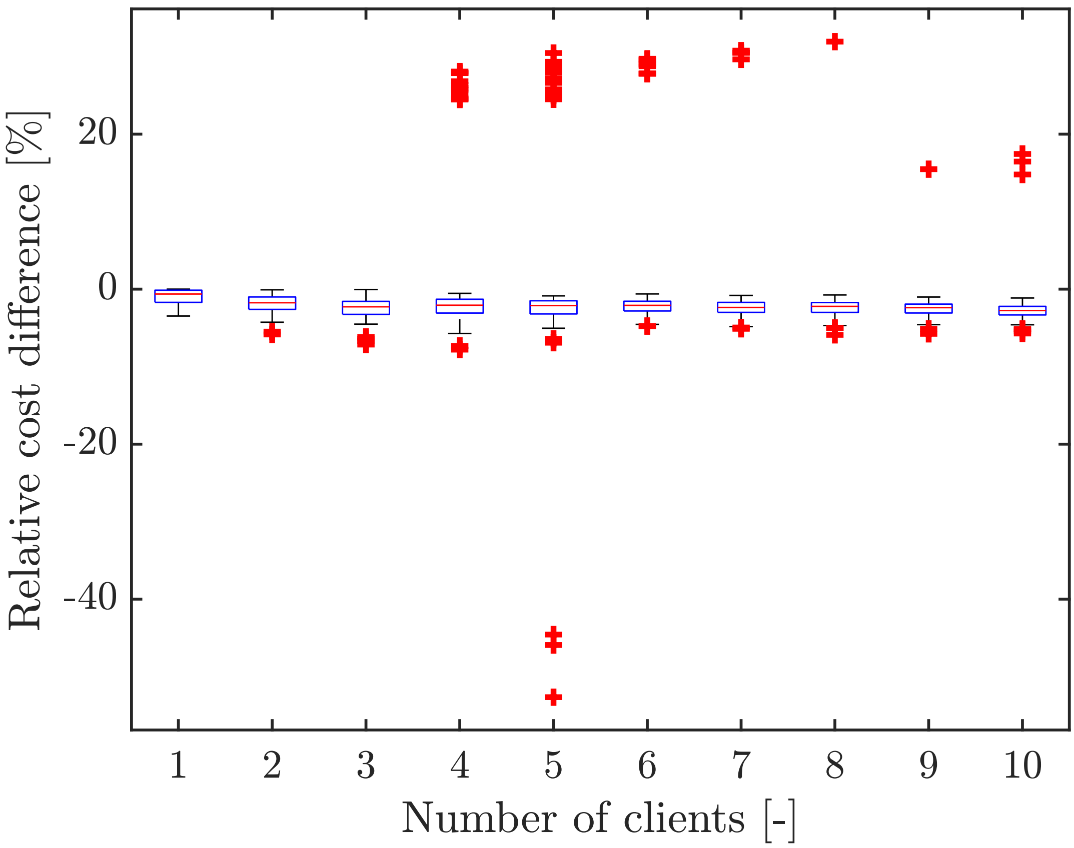



Figure B.2 – Boxplots of the relative cost difference between ATM and ADM in Belgium

# List of Figures

# List of Tables

# References


ACEA. (2020). https://www.acea.be/press-releases/article/fuel-types-of-new-trucks-diesel-97.9-electric-0.2-hybrid-0.1-market-share-i.

Bellman, R. (1954, 11). The theory of dynamic programming. *Bull. Amer. Math. Soc.*, *60*(6), 503–515. Retrieved from https://projecteuclid.org:443/euclid.bams/1183519147

Bellman, R. (1962). Dynamic Programming Treatment of the Travelling Salesman Problem. *Journal of the Association for Computing Machinery,*, *9*(1), 61–63.

CGSLB. (2014). *Cp 140.03 : conditions de travail et de rémunération.* https://www.cgslb.be/fr/cp-14003-conditions-de-travail-et-de-remuneration#indemnits-primes-et-avantages.

Comité National Routier. (2020a). *Camion-remorque porte-voitures 35 t.* http://www.cnr.fr/Indices-Statistiques/Camion-remorque-porte-voitures-35-T/Referentiel-prix-de-revient.

Comité National Routier. (2020b). *Longue distance ea.* http://www.cnr.fr/Indices-Statistiques/Longue-distance-EA/Referentiel-prix-de-revient.

European Commission. (2018). On the road to automated mobility: An EU strategy for mobility of the future EN.

European Commission. (2020). Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL establishing the framework for achieving climate neutrality and amending Regulation (EU) 2018/1999 (European Climate Law).

European Commmission. (2019). *Transport taxes and charges in europe: An overview study of economic internalisation measures applied in europe.* https://ec.europa.eu/transport/sites/transport/files/studies/transport-taxes-and-charges-in-europe-isbn-978-92-79-99561-3.pdf.

Funke, B., Grünert, T., & Irnich, S. (2005). Local search for vehicle routing and scheduling problems: Review and conceptual integration. *Journal of Heuristics*, *11*(4), 267–306. doi: 10.1007/s10732-005-1997-2

Goel, A., & Gruhn, V. (2005, 01). Solving a dynamic real-life vehicle routing problem. In (Vol. 2005, p. 367-372). doi: 10.1007/3-540-32539-5_58

Goel, A., & Gruhn, V. (2006). Drivers' working hours in vehicle routing and scheduling. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*(October 2006), 1280–1285. doi: 10.1109/itsc.2006.1707399

Google. (2020). https://cloud.google.com/maps-platform/pricing/?hl=fr.

Government of the Netherlands. (2020). *Plans for introduction of hgv toll.* https://www.government.nl/topics/freight-transportation/plans-for-introduction-of-hgv-toll.

GraphHopper. (2020). https://www.graphhopper.com/.

Held, M., & Karp, R. M. (1962). A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, *10*(1), 196–210. Retrieved from http://www.jstor.org/stable/2098806

ICCT. (2018). *Fuel consumption testing of tractortrailers in the european union and the united states.* https://theicct.org/sites/default/files/publications/EU_HDV_Testing_BriefingPaper_20180515a.pdf.

ITLB. (2015). *Analyse économique du transport routier de marchandises en belgique.* http://94.23.228.57/ITLB_WEB/Documents/fr/Analyse%20Economique/Analyse_eco_fr_3.pdf.

Karp, R. M. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, 85–103.

Kok, A. L., Meyer, C. M., Kopfer, H., & Schutten, J. M. (2010). A dynamic programming heuristic for the vehicle routing problem with time windows and European community social legislation. *Transportation Science*, *44*(4), 442–454. doi: 10.1287/trsc.1100.0331

Kopfer, H., & Meyer, C. M. (2009). A Model for the Traveling Salesman Problem Including the EC Regulations on Driving Hours. *Operations Research Proceedings 2008*. doi: 10.1007/978-3-642-00142-0

Lambert, J. (2019). *Les camions autonomes, une révolution imminente dans le secteur routier belge ?* (Unpublished doctoral dissertation). University of Liège.

Mappy. (2020). https://blog.mappy.com/entreprise/nos-offres-commerciales/.

Miller, C., Tucker, A., & Zemlin, R. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, *7*.

Parliament, E., & Council, E. (n.d.-a). Directive 2002/15/ec of the european parliament and of the council. *Official Journal of the European Communities*.

Parliament, E., & Council, E. (n.d.-b). Regulation (ec) no 561/2006 of the european parliament and of the council.

Raposo, A., Grosso, M., Macías, F., Galassi, E., Krasenbrink, C., Krause, A., … Ciuffo, C. (2018). *An analysis of possible socio-economic effects of a Cooperative, Connected and Automated Mobility (CCAM) in Europe: Effects of automated driving on the economy, employment and skills*. Retrieved from https://ec.europa.eu/jrc doi: 10.2760/777

Ryan, M. (2020). The Future of Transportation: Ethical, Legal, Social and Economic Impacts of Self-driving Vehicles in the Year 2025. *Science and Engineering Ethics*, *26*(3), 1185–1208. Retrieved from https://doi.org/10.1007/s11948-019-00130-2 doi: 10.1007/s11948-019-00130-2

SPW Fiscalité. (2020). *Montants de la taxe de circulation - véhicules destinés au transport de marchandises.* https://www.wallonie.be/sites/default/files/2019-05/baremes_taxe_de_circulation_camions_remorques.pdf.

Transport & Environment. (2015). https://www.transportenvironment.org/sites/te/files/publications/2015%2009%20TE%20Briefing%20Truck%20CO2%20Too%

20big%20to%20ignore_FINAL.pdf.
Union, E. (n.d.-a). https://ec.europa.eu/transport/modes/road/social_provisions/driving_time_en.
Union, E. (n.d.-b). https://ec.europa.eu/transport/modes/road/social_provisions/working_time_en.
United Nations. (1968). Convention on Road Traffic.
Van Hoorn, J. J. (2016). *Dynamic Programming for Routing and Scheduling: Optimizing Sequences of Decisions*. Retrieved from https://research.vu.nl/en/publications/dynamic-programming-for-routing-and-scheduling-optimizing-sequenc
van de Camp, W. (2018). on autonomous driving in European transport.
Viappas. (2020). https://www.viapass.be/wp-content/uploads/2020/06/072020-tarieven-fr.pdf.

# Contents

The world is on the edge of an economical breakthrough with the emergence of autonomous cars. Such vehicles do not, or for a very short time, require the driver's attention or even not require a driver at all. This new era of vehicles is largely promoted by Tesla's eccentric CEO Elon Musk and the promotion of the S 3 X Y series of vehicles which include an autopilot mode. In parallel to the emergence of autonomous cars, autonomous trucks are also in development and are likely to revolutionize the transport sector and in particular freight transport. Again, Tesla wants to be a pioneer in this domain with the Tesla Semi, which is supposed to arrive around 2021 on the market. Autonomous vehicles are expected to improve road safety by drastically reducing the number of incidents. Furthermore, they are likely to be an excellent ally in the fight against climate change by improving the road transport efficiency and reducing greenhouse gas emissions.

The aim of this work is to emphasize the gains of using autonomous trucks over regular trucks for freight transport. These gains are mainly based on the prime costs of such trucks but also, in a lesser extent, to the greenhouse gas emissions. To do so, two similar trucks, one with a driver and an autonomous one, are compared for different delivery situations, from national to international level. While the autonomous truck can drive continuously, the driver is submitted to the European legislation on driving and working hours. Their respective journey, which visits a set of $n$ clients selected randomly, is computed by minimizing either the driving time or the driving distance. To each route, a monetary cost is associated which serves as a basis for the comparison between the two trucks.

***Keywords:*** Autonomous trucks, Vehicle Routing Problem, Optimization